aws

Guía del usuario

Amazon ElastiCache

Versión de API 2015-02-02

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon ElastiCache: Guía del usuario

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

¿Qué es ElastiCache? 1
Almacenamiento en caché sin servidor 1
Clústeres basados en nodos 2
Servicios relacionados 2
Funcionamiento 3
Caché y motores de almacenamiento en caché 3
Cómo elegir una opción de implementación 9
Recursos de ElastiCache 18
AWSRegiones y zonas de disponibilidad de 19
Casos de uso 21
Introducción a ElastiCache 33
Configuración de ElastiCache 33
Cómo crear una Cuenta de AWS 33
Creación de un usuario con acceso administrativo 34
Conceder acceso programático 35
Configuración de permisos 37
Configuración de EC2 38
Otorgar acceso de red 38
Configuración del acceso a la línea de comandos 39
Creación de una caché sin servidor de Valkey 40
Lectura y escritura de datos 42
Limpieza 43
Siguientes pasos 44
Creación de una caché sin servidor de Redis OSS 45
Lectura y escritura de datos 46
Limpieza 48
Siguientes pasos 49
Creación de una caché sin servidor de Memcached 49
Lectura y escritura de datos 50
Limpieza 55
Siguientes pasos 56
Tutoriales: introducción a Python y ElastiCache 56
Python y ElastiCache 57
Tutorial: configuración de Lambda para acceder a ElastiCache en una VPC 75

Paso 1: creación de una caché sin servidor de ElastiCache. 75
Paso 2: creación de una función de Lambda para ElastiCache 78
Paso 3: prueba de la función de Lambda con ElastiCache 82
Paso 4: limpieza (opcional) 83
Creación y administración de un clúster de ElastiCache basado en nodos 85
Componentes y características 85
Nodos 86
Particiones de ElastiCache 87
Clústeres de ElastiCache 87
Replicación de ElastiCache 89
Puntos de conexión de ElastiCache 92
Grupos de parámetros 93
Seguridad de ElastiCache 93
Grupos de subredes 94
Copias de seguridad de ElastiCache 94
Eventos 95
Terminología de ElastiCache 96
Tutorial: Cómo crear un clúster de ElastiCache basado en nodos 98
Creación de un clúster de ElastiCache basado en nodos para Valkey 98
Creación de un clúster de ElastiCache basado en nodos para Redis OSS 120
Eliminación de un clúster 142
Otros tutoriales y vídeos 144
Videos 145
Administración de nodos en ElastiCache 150
Visualización del estado de los nodos en ElastiCache 151
Nodos y particiones de Valkey o Redis OSS 158
Conexión a los nodos 160
Tipos de nodos compatibles 165
Reinicio de nodos 181
Reemplazo de nodos (Valkey y Redis OSS) 186
Reemplazo de nodos (Memcached) 194
Nodos reservados 196
Migración de nodos de generación anterior 212
Administración de clústeres en ElastiCache 216
Elección de un tipo de red en ElastiCache 218
Detección automática (Memcached) 223

Organización de datos en niveles en ElastiCache 267
Preparación de un clúster en ElastiCache 274
Creación de un clúster para Valkey o Redis OSS 285
Creación de un clúster para Memcached 295
Visualización de los detalles de un clúster de ElastiCache 299
Modificación de un clúster de ElastiCache 313
Adición de nodos a un clúster de ElastiCache 319
Eliminación de nodos de un clúster de ElastiCache 330
Cancelación de operaciones pendientes de adición o eliminación de nodos en ElastiCache 339
Eliminación de un clúster en ElastiCache 340
Acceso al clúster o grupo de replicación de ElastiCache 343
Búsqueda de puntos de conexión en ElastiCache 352
Particiones en ElastiCache 368
Comparación de los clústeres Valkey, Memcached y Redis OSS basados en nodos 374
Migración en línea para Valkey o Redis OSS 379
Descripción general 380
Pasos para realizar la migración 381
Preparación del origen y destino para la migración 381
Prueba de la migración de datos 383
Inicio de la migración 384
Verificación del progreso de migración de datos 385
Finalización de la migración de datos 386
Realización de la migración de datos en línea con la consola 387
Elección de regiones y zonas de disponibilidad para ElastiCache 388
Consideraciones sobre las zonas de disponibilidad con Memcached 389
Ubicación de los nodos 392
Regiones y puntos de conexión admitidos 392
Uso de zonas locales con ElastiCache 397
Uso de Outposts con ElastiCache 399
Uso de ElastiCache 404
Instantánea y restauración 404
Restricciones 405
Impacto en el rendimiento de las copias de seguridad de los clústeres basados en nodos .. 406
Programación de copias de seguridad automáticas 408
Copias de seguridad manuales 410

Creación de una copia de seguridad final ... 417
Descripción de copias de seguridad ... 420
Copiar copias de seguridad ... 422
Exportación de una copia de seguridad ... 425
Restauración desde una copia de seguridad ... 434
Eliminación de una copia de seguridad ... 435
Etiquetado de copias de seguridad ... 437
Tutorial: propagación de un clúster basado en nodos con una copia de seguridad ... 438
Versiones del motor y actualización en ElastiCache ... 448
Actualización de las versiones del motor ... 448
Soporte extendido ... 453
Administración de versiones para ElastiCache ... 457
Valley 8.2 ... 464
Valley 8.1 ... 464
Valley 8.0 ... 465
ElastiCache versión 7.2.6 para Valkey ... 466
Versiones compatibles de Redis OSS ... 466
Programa de fin de vida útil de las versiones de Redis OSS ... 481
Versiones compatibles de Memcached ... 483
Diferencias de compatibilidad y comportamiento de versiones principales del motor con Valkey ... 487
Diferencias de compatibilidad y comportamiento de las versiones principales del motor con Redis OSS ... 487
Consideraciones sobre la actualización al trabajar con clústeres basados en nodos ... 492
Procedimientos recomendados y estrategias de almacenamiento en caché ... 495
Prácticas recomendadas generales ... 495
Prácticas recomendadas para utilizar réplicas de lectura ... 496
Comandos de Valkey, Memcached y Redis OSS compatibles y restringidos ... 500
Configuración y límites de Valkey y Redis OSS ... 539
Ejemplos de clientes de IPv6 para Valkey, Memcached y Redis OSS ... 542
Prácticas recomendadas para clientes (Valkey y Redis OSS) ... 543
Prácticas recomendadas para clientes (Memcached) ... 567
Clústeres de ElastiCache de doble pila compatibles con TLS ... 572
Administración de la memoria reservada para Valkey y Redis OSS ... 575
Prácticas recomendadas al trabajar con clústeres basados en nodos de Valkey y Redis OSS ... 582

Estrategias de almacenamiento en caché para Memcached 589
Administración del clúster basado en nodos en ElastiCache 595
Clústeres de escalado automático de Valkey y Redis OSS 595
Modificación del modo de clúster 645
Replicación en regiones de AWS mediante los almacenes de datos globales 648
Alta disponibilidad a través de grupos de reproducción 677
Administración del mantenimiento de clústeres de ElastiCache 768
Configuración de los parámetros de motor mediante grupos de parámetros de ElastiCache 771
Conexión automática de una instancia de EC2 y una caché de ElastiCache 893
Conectividad de EC2 automática 894
Visualización de los recursos de computación conectados 900
Escalado de ElastiCache 901
Escalado de ElastiCache Serverless 901
Establecimiento de límites de escalado para administrar los costes 901
Escalado previo con ElastiCache sin servidor 901
Establecimiento de límites de escalado mediante la consola y la AWS CLI 903
Escalado de clústeres basados en nodos 904
Introducción a los filtros de Bloom 995
Tipo de datos Bloom 995
Límite del tamaño de Bloom 996
ACL 997
Métricas relacionadas con Bloom 997
Comandos de filtros de Bloom 998
Introducción a la búsqueda vectorial 999
Información general de la búsqueda vectorial 1000
Características y límites de la búsqueda vectorial 1004
Selección de la configuración adecuada 1007
Comandos de búsqueda vectorial 1008
Introducción a JSON para Valkey y Redis OSS 1021
Información general del tipo de datos JSON 1022
Comandos JSON 1034
Etiquetado de los recursos de ElastiCache 1077
Monitoreo de costos con etiquetas 1090
Administrar etiquetas con AWS CLI 1091
Administración de etiquetas mediante la API de ElastiCache 1096

Enfoque Well-Architected de Amazon ElastiCache 1099
Pilar de excelencia operativa 1100
Pilar de seguridad 1110
Pilar de fiabilidad 1116
Pilar de eficiencia de rendimiento 1123
Pilar de optimización de costos 1134
Solución de problemas en ElastiCache 1141
Problemas de conectividad 1142
Errores del cliente de Valkey o Redis OSS 1142
Solución de problemas de alta latencia en ElastiCache sin servidor 1143
Solución de problemas de limitación en ElastiCache sin servidor 1145
Problemas de las conexiones persistentes 1146
Temas relacionados 1168
Seguridad 1169
Protección de los datos 1170
Seguridad de datos en Amazon ElastiCache 1171
Privacidad del tráfico entre redes 1250
Seguridad de ElastiCache y Amazon VPC 1250
Puntos de conexión de VPC de interfaz y API de ElastiCache (AWS PrivateLink) 1277
Subredes y grupos de subredes 1283
Identity and Access Management 1291
Público 1291
Autenticación con identidades 1292
Administración de acceso mediante políticas 1293
Cómo funciona Amazon ElastiCache con IAM 1295
Ejemplos de políticas basadas en identidades 1301
Solución de problemas 1304
Control de acceso 1306
Información general sobre la administración del acceso 1307
Validación de conformidad 1359
Más información 1360
Resiliencia 1360
Mitigación de errores 1361
Seguridad de la infraestructura 1366
Actualizaciones de servicio 1366
Administración de actualizaciones de servicio 1367

Vulnerabilidad de seguridad corregida 1373
Registro y supervisión 1375
Métricas y eventos para Valkey y Redis OSS sin servidor 1375
Métricas sin servidor 1375
Eventos sin servidor 1385
Métricas y eventos para clústeres basados en nodos de Valkey y Redis OSS 1400
Métricas de clústeres basados en nodos 1400
Eventos para clústeres basados en nodos (Valkey y Redis OSS) 1400
Eventos y métricas para Memcached 1409
Métricas sin servidor de Memcached 1409
Eventos sin servidor de Memcached 1413
Registro de llamadas a la API de Amazon ElastiCache con AWS CloudTrail 1424
Información sobre Amazon ElastiCache en CloudTrail 1425
Descripción de las entradas de archivo de registros de Amazon ElastiCache 1426
Supervisión de eventos en Amazon SNS 1430
Administración de notificaciones de Amazon SNS de ElastiCache 1430
Visualización de eventos de ElastiCache 1435
Notificaciones de eventos y Amazon SNS 1439
Entrega de registro 1447
Contenido de una entrada de registro lento 1448
Contenido de una entrada de registro del motor 1448
Permisos para configurar el registro 1448
Especificaciones del tipo y formato de registro 1449
Destinos de registro de ElastiCache 1450
Especificación de la entrega de registros mediante la consola 1453
Especificación de la entrega de registros mediante la AWS CLI 1454
Monitoreo del uso 1460
Métricas de nivel de host 1460
Métricas de Valkey y Redis OSS 1464
Métricas de Memcached 1483
¿Qué métricas debo monitorear? 1489
Elección de periodos y estadísticas de métricas 1494
Monitoreo de métricas de nodo y clúster de CloudWatch 1494
Cuotas 1498
Referencia 1501
Uso de la API de ElastiCache 1501

Uso de la API de consultas 1501
Bibliotecas disponibles 1505
Solución de problemas de aplicaciones 1505
Configurar la AWS CLI de ElastiCache 1506
Requisitos previos 1507
Obtención de las herramientas de línea de comandos 1509
Configuración de las herramientas 1509
Suministro de credenciales para las herramientas 1510
Variables de entorno 1511
Mensajes de error 1512
Notificaciones 1514
Notificaciones generales de ElastiCache 1514
Notificaciones de ElastiCache for Memcached 1514
Notificaciones específicas de ElastiCache para Redis OSS 1515
Historial de la documentación de ElastiCache 1516
AWSGlosario de 1560

¿Qué es Amazon ElastiCache?

Le damos la bienvenida a la Guía del usuario de Amazon ElastiCache. Amazon ElastiCache es un servicio web que facilita la configuración, la administración y el escalado de un almacén de datos en memoria distribuido o de un entorno de caché en la nube: Proporciona una capacidad de almacenamiento en caché de alto rendimiento, escalable y rentable. Al mismo tiempo, ayuda a eliminar la complejidad propia de la implementación y la administración de un entorno de caché distribuido.

Puede utilizar Amazon ElastiCache en dos formatos. Puede empezar con una memoria caché sin servidor o crear un clúster basado en nodos.

> **Note**
>
> Amazon ElastiCache es compatible con los motores de Valkey, Memcached y Redis OSS. Si no está seguro de qué motor desea utilizar, consulte Comparación de los clústeres Valkey, Memcached y Redis OSS basados en nodos en esta guía.

Almacenamiento en caché sin servidor

ElastiCache ofrece almacenamiento en caché sin servidor, lo que simplifica la adición y el funcionamiento de una caché para la aplicación. ElastiCache Serverless le permite crear una caché de alta disponibilidad en menos de un minuto y elimina la necesidad de aprovisionar instancias o configurar nodos o clústeres. Los desarrolladores pueden crear una caché sin servidor especificando el nombre de la caché mediante la consola ElastiCache, el SDK o la CLI.

ElastiCache Serverless también elimina la necesidad de planificar y administrar la capacidad de almacenamiento en caché. ElastiCache supervisa constantemente la memoria caché, el procesamiento y el ancho de banda de la red que utiliza la aplicación, y se escala para satisfacer las necesidades de la aplicación. ElastiCache ofrece una experiencia de punto de conexión sencilla para los desarrolladores, ya que abstrae la infraestructura de caché subyacente y el diseño del clúster. ElastiCache gestiona el aprovisionamiento de hardware, la supervisión, las sustituciones de nodos y los parches de software de forma automática y transparente, para que pueda centrarse en el desarrollo de aplicaciones, y no en el uso de la memoria caché.

ElastiCache sin servidor es compatible con Valkey 7.2, Memcached 1.6.21 y versiones posteriores, y Redis OSS 7.1 y versiones posteriores.

Creación de un clúster basado en nodos

Si necesita un control detallado del clúster de ElastiCache, puede optar por crear un clúster de Valkey, Memcached o Redis OSS basado en nodos. ElastiCache le permite crear un clúster basado en nodos eligiendo el tipo de nodo, el número de nodos y la ubicación de los nodos en las zonas de disponibilidad de AWS para el clúster. Como ElastiCache es un servicio totalmente administrado, gestiona automáticamente el aprovisionamiento de hardware, la supervisión, las sustituciones de nodos y la aplicación de parches de software para el clúster.

Crear un clúster basado en nodos le aporta mayor flexibilidad y control sobre los clústeres. Por ejemplo, puede operar un clúster con disponibilidad en una o muchas zonas de disponibilidad, en función de sus necesidades. También puede ejecutar Valkey, Memcached o Redis OSS en modo de clúster para habilitar el escalado horizontal o sin modo de clúster si solo se escala verticalmente. Al crear un clúster basado en nodos, usted será el responsable de elegir el tipo y la cantidad de nodos correctamente para garantizar que la caché tenga la capacidad suficiente según lo requiera la aplicación. También puede elegir cuándo aplicar los nuevos parches de software a su clúster de Valkey o Redis OSS.

Al crear un clúster basado en nodos, puede elegir entre varias versiones compatibles de Valkey, Memcached y Redis OSS. Para obtener más información acerca de las versiones de motor admitidas, consulte Versiones del motor y actualización en ElastiCache.

Servicios relacionados

MemoryDB

Al decidir si utilizar ElastiCache o MemoryDB, tenga en cuenta las siguientes comparaciones:

- ElastiCache es un servicio que se utiliza normalmente para almacenar en caché datos de otras bases de datos y almacenes de datos con Valkey, Memcached o Redis OSS. Debería considerar ElastiCache para almacenar en caché cargas de trabajo en las que desee acelerar el acceso a los datos con la base de datos principal o almacén de datos existente (rendimiento de lectura y escritura en microsegundos). También debe considerar ElastiCache para los casos de uso en los que desee utilizar las estructuras de datos y las API de Valkey o Redis OSS para acceder a los datos almacenados en una base de datos o un almacén de datos principal.
- ElastiCache puede ayudarle también a ahorrar costos de bases de datos al almacenar en caché los datos a los que se accede con frecuencia. Si su aplicación tiene requisitos de alto rendimiento de lectura, puede lograr una alta escalabilidad, un rendimiento rápido y una reducción de los

costos de almacenamiento de datos mediante ElastiCache, en lugar de escalar la base de datos subyacente.

- MemoryDB es una base de datos en memoria duradera para cargas de trabajo que requieren una base de datos principal ultrarrápida. Es compatible con Valkey y Redis OSS. Debería considerar el uso de MemoryDB si la carga de trabajo requiere una base de datos duradera que ofrezca un rendimiento ultrarrápido (lectura en microsegundos y latencia de escritura de un solo dígito en milisegundos). MemoryDB también puede ser una buena opción para el caso de uso si desea crear una aplicación mediante las estructuras de datos y las API de Valkey o Redis OSS con una base de datos principal y duradera. Por último, debería considerar el uso de MemoryDB para simplificar la arquitectura de la aplicación y reducir los costos al sustituir el uso de una base de datos por una memoria caché para aumentar la durabilidad y el rendimiento.

Amazon Relational Database Service

ElastiCache puede ayudarle a ahorrar costos de bases de datos al almacenar en caché los datos a los que se accede con frecuencia. Si su aplicación tiene requisitos de alto rendimiento de lectura, puede lograr una alta escalabilidad, un rendimiento rápido y una reducción de los costos de almacenamiento de datos mediante ElastiCache, en lugar de escalar la base de datos subyacente.

Para obtener más información general sobre el servicio Amazon Relational Database Service, consulte Amazon RDS.

ElastiCache puede ayudarle a ahorrar costos de bases de datos al almacenar en caché los datos a los que se accede con frecuencia. Si su aplicación tiene requisitos de alto rendimiento de lectura, puede lograr una alta escalabilidad, un rendimiento rápido y una reducción de los costos de almacenamiento de datos mediante ElastiCache, en lugar de escalar la base de datos subyacente.

Cómo funciona ElastiCache

Aquí encontrará información general sobre los componentes principales de una implementación de ElastiCache.

Caché y motores de almacenamiento en caché

Una caché es un almacén de datos en memoria que puede utilizar para almacenar datos en caché. Por lo general, la aplicación almacenará en caché los datos a los que se accede con frecuencia para optimizar los tiempos de respuesta. ElastiCache ofrece dos opciones de implementación: caché sin servidor y clústeres basados en nodos. Consulte Cómo elegir una opción de implementación.

> **Note**
>
> Amazon ElastiCache es compatible con los motores de Valkey, Memcached y Redis OSS. Si no está seguro de qué motor desea utilizar, consulte Comparación de los clústeres Valkey, Memcached y Redis OSS basados en nodos en esta guía.

Temas

- Cómo funciona ElastiCache
- Dimensiones de precios
- Copias de seguridad de ElastiCache

Cómo funciona ElastiCache

ElastiCache Serverless

ElastiCache Serverless le permite crear una memoria caché sin preocuparse por la planificación de la capacidad, la administración del hardware o el diseño del clúster. Solo tiene que proporcionar un nombre para la caché y recibirá un único punto de conexión que podrá configurar en su cliente de Valkey, Memcached y Redis OSS para empezar a acceder a la caché.

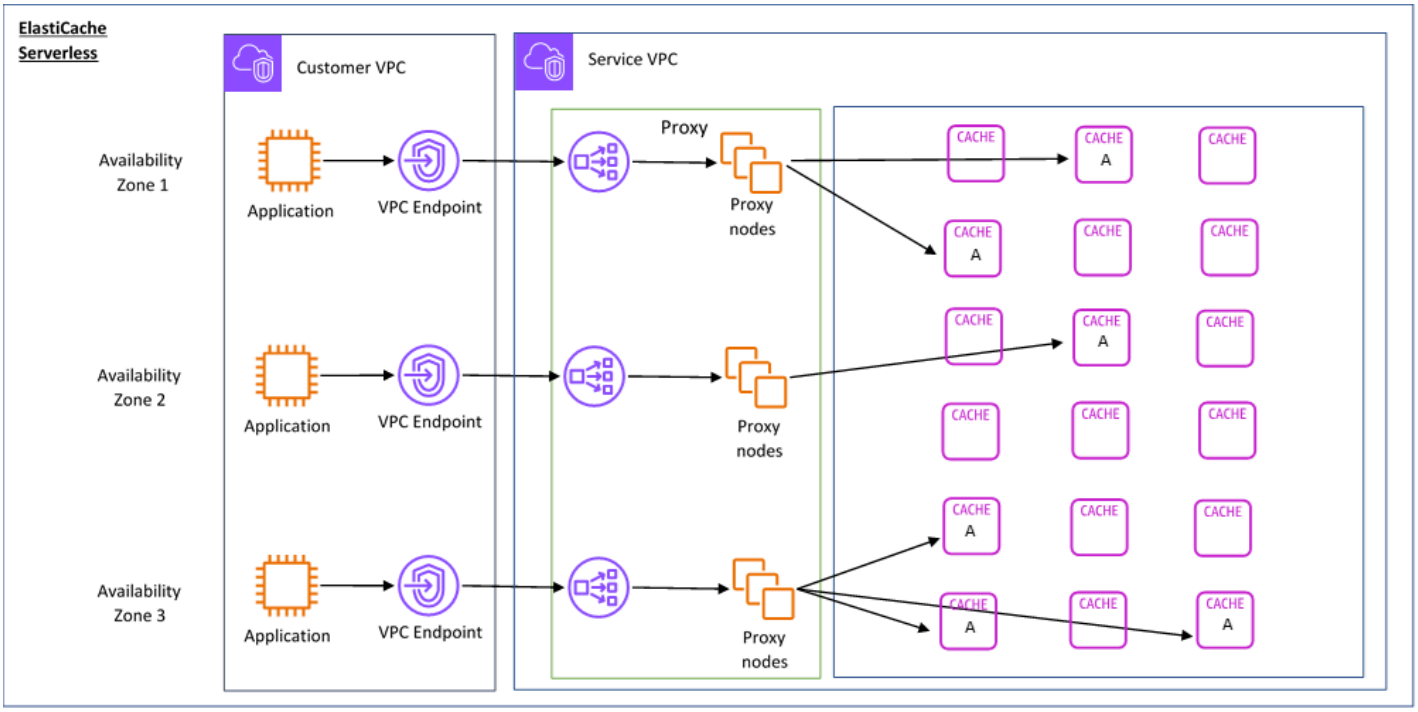
> **Note**
>
> - ElastiCache sin servidor ejecuta Valkey, Memcached o Redis OSS en modo de clúster y solo es compatible con los clientes que admiten TLS.

Ventajas principales

- No hay planificación de la capacidad: con ElastiCache Serverless, no es necesario planificar la capacidad. ElastiCache Serverless supervisa de forma continua el uso de memoria, procesamiento y ancho de banda de la caché y escala tanto vertical como horizontalmente. Permite que un nodo de caché aumente de tamaño y, en paralelo, inicia una operación de escalado horizontal para garantizar que la caché pueda escalarse a fin de cumplir con los requisitos de la aplicación en todo momento.

- Pago por uso: con ElastiCache Serverless, pagará por los datos almacenados y el procesamiento empleado por su carga de trabajo en la memoria caché. Consulte Dimensiones de precios.
- Alta disponibilidad: ElastiCache Serverless replica automáticamente los datos en varias zonas de disponibilidad (AZ) para lograr una alta disponibilidad. Supervisa automáticamente los nodos de caché subyacentes y los reemplaza en caso de errores. Ofrece un SLA de disponibilidad del 99,99 % para cada caché.
- Actualizaciones de software automáticas: ElastiCache Serverless actualiza automáticamente la memoria caché a la última versión secundaria y parchea el software sin que ello afecte a la disponibilidad de la aplicación. Cuando haya disponible una nueva versión principal, ElastiCache le enviará una notificación.
- Seguridad: la tecnología sin servidor cifra siempre los datos en tránsito y en reposo. Puede utilizar una clave administrada por el servicio o su propia clave administrada por el cliente para cifrar los datos en reposo.

El siguiente diagrama ilustra el funcionamiento de ElastiCache Serverless.



Al crear una nueva caché sin servidor, ElastiCache crea un punto de conexión de nube privada virtual (VPC) en las subredes que elija de la VPC. La aplicación se puede conectar a la memoria caché a través de estos puntos de conexión de VPC.

Con ElastiCache Serverless, recibirá un único punto de conexión DNS al que se conecta su aplicación. Cuando solicita una nueva conexión al punto de conexión, ElastiCache Serverless gestiona todas las conexiones de caché a través de una capa de proxy. La capa de proxy contribuye a reducir la compleja configuración del cliente, ya que este no necesita volver a detectar de nuevo la topología del clúster en caso de que haya cambios en el clúster subyacente. La capa proxy es un conjunto de nodos proxy que gestionan las conexiones mediante un equilibrador de carga de red.

Cuando la aplicación crea una nueva conexión de caché, el equilibrador de carga de red envía la solicitud a un nodo proxy. Cuando la aplicación ejecuta comandos de caché, el nodo proxy que está conectado a la aplicación ejecuta las solicitudes en un nodo de caché de la memoria caché. La capa proxy abstrae la topología del clúster y los nodos del cliente. Gracias a esto, ElastiCache puede hacer las siguientes tareas de un modo inteligente: equilibrar la carga, escalar horizontalmente y añadir nuevos nodos de caché, reemplazar los nodos de caché cuando fallan y actualizar el software en los nodos de caché; todo ello, sin que la aplicación deje de estar disponible y sin tener que restablecer las conexiones.

Clústeres basados en nodos

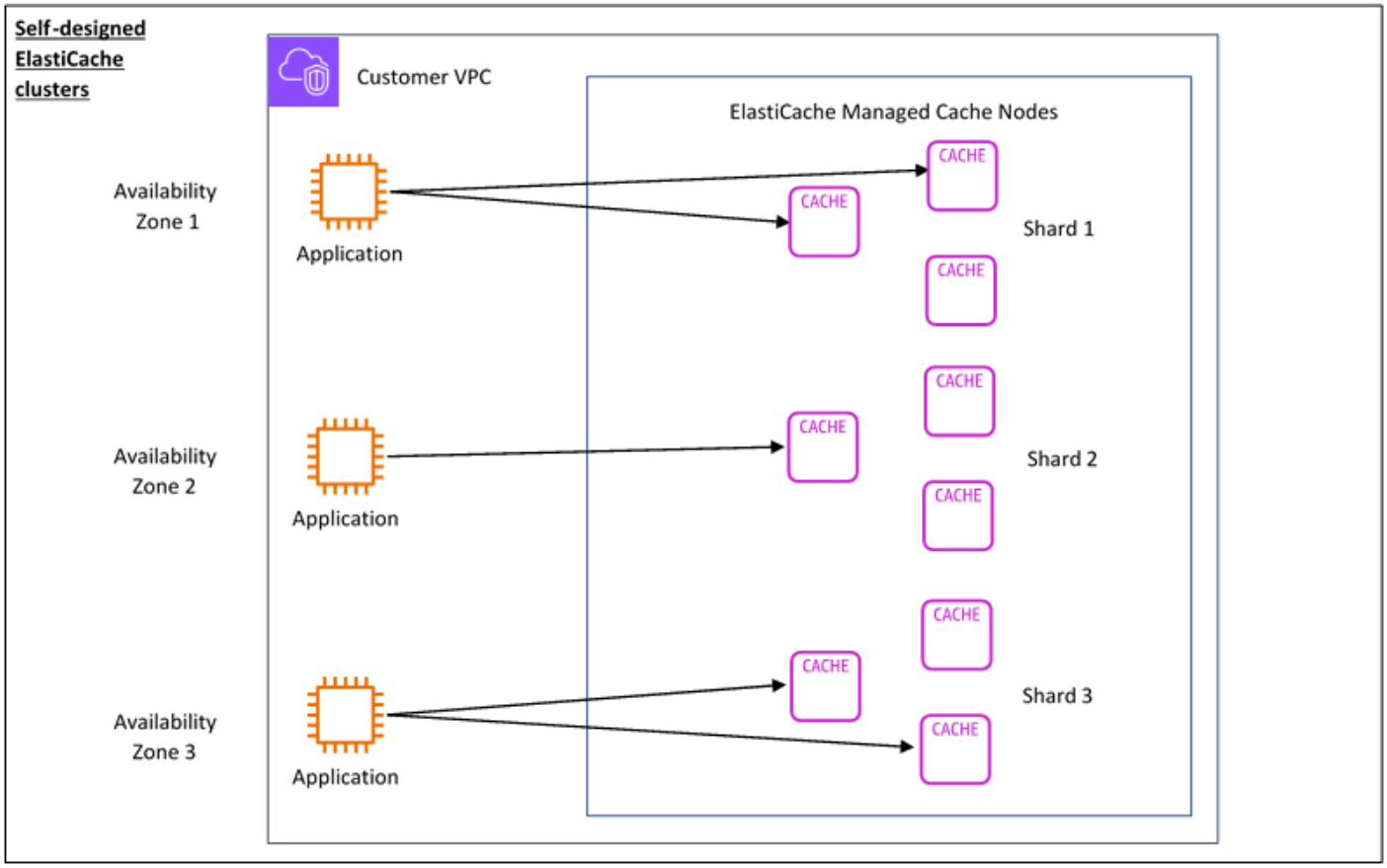
Puede crear un clúster de ElastiCache basado en nodos seleccionando una familia de nodos de caché, un tamaño y un número de nodos para el clúster. Crear un clúster basado en nodos le proporciona un control más detallado y le permite elegir la cantidad de particiones de la memoria caché y el número de nodos (principales y réplicas) de cada partición. Puede utilizar Valkey o Redis OSS en modo de clúster creando un clúster con varias particiones o en modo sin clúster con una sola partición.

Ventajas principales

- Crear un clúster basado en nodos: con ElastiCache, puede crear un clúster basado en nodos y elegir dónde quiere colocar los nodos de caché. Por ejemplo, si tiene una aplicación en la que quiere sacrificar una alta disponibilidad por una baja latencia, puede optar por implementar los nodos de caché en una única zona de disponibilidad. Si lo prefiere, puede crear un clúster basado en nodos con nodos en varias zonas de disponibilidad para lograr una alta disponibilidad.
- Control detallado: al crear un clúster basado en nodos, tenemos más control a la hora de refinar la configuración de la caché. Por ejemplo, puede utilizar Parámetros de Valkey y Redis OSS y Parámetros específicos de Memcached para configurar el motor de la caché.
- Escalar vertical y horizontalmente: puede escalar el clúster manualmente aumentando o disminuyendo el tamaño del nodo de caché cuando sea necesario. Además, puede escalar horizontalmente añadiendo particiones nuevas o añadiendo más réplicas a sus particiones.

También puede usar la característica de escalado automático para configurar el escalado en función de un cronograma o el escalado en función de métricas como el uso de CPU y memoria en la caché.

El siguiente diagrama ilustra el funcionamiento de los clústeres basados en nodos de ElastiCache.



Dimensiones de precios

Hay dos opciones de implementación para ElastiCache. Al implementar ElastiCache Serverless, pagará por el uso de los datos almacenados (en GB-hora) y el procesamiento (en unidades de procesamiento de ElastiCache o ECPU). Al crear un clúster basado en nodos, se paga por hora de uso del nodo de caché. Puede encontrar más información sobre los precios aquí.

Almacenamiento de datos

Usted paga por los datos almacenados en ElastiCache Serverless facturados en gigabytes-hora (GB-hora). ElastiCache Serverless supervisa de forma continua los datos almacenados en la memoria

caché, realiza muestreos varias veces por minuto y calcula un promedio por hora para determinar el uso de almacenamiento de datos de la memoria caché en GB-hora. Cada caché de ElastiCache Serverless se mide para un mínimo de 1 GB de datos almacenados.

Unidades de procesamiento de ElastiCache (ECPU)

Usted paga por las solicitudes que su aplicación ejecuta en ElastiCache Serverless en unidades de procesamiento (ECPU) de ElastiCache, una unidad que incluye tanto el tiempo de la vCPU como los datos transferidos.

- Las lecturas y escrituras sencillas requieren 1 ECPU por cada kilobyte (KB) de datos transferidos. Por ejemplo, un comando GET que transfiere hasta 1 KB de datos consume 1 ECPU. Una solicitud SET que transfiera 3,2 KB de datos consumirá 3,2 ECPU.
- Con Valkey y Redis OSS, los comandos que consumen más tiempo de vCPU y transfieren más datos consumen ECPU en función del valor más alto entre las dos dimensiones. Por ejemplo, si la aplicación usa el comando HMGET, consume el triple de tiempo de vCPU que un simple comando SET/GET y tranfiere 3,2 KB de datos, consumirá 3,2 ECPU. Sin embargo, si transfiere solo 2 KB de datos, consumirá 3 ECPU.
- Con Valkey y Redis OSS, los comandos que requieren más tiempo de vCPU consumirán proporcionalmente más ECPU. Por ejemplo, si la aplicación usa el comando HMGET de Valkey o Redis OSS y consume el triple de tiempo de vCPU que un simple comando SET/GET, consumirá 3 ECPU.
- Con Memcached, los comandos que funcionan con varios elementos consumirán proporcionalmente más ECPU. Por ejemplo, si la aplicación realiza una multiobtención en tres elementos, consumirá 3 ECPU.
- Con Memcached, los comandos que operan en más elementos y transfieren más datos consumen ECPU en función del valor más alto de las dos dimensiones. Por ejemplo, si la aplicación utiliza el comando GET, recupera tres elementos y transfiere 3,2 KB de datos, consumirá 3,2 ECPU. Sin embargo, si transfiere solo 2 KB de datos, consumirá 3 ECPU.

ElastiCache Serverless emite una nueva métrica denominada `ElastiCacheProcessingUnits` que le ayuda a comprender las ECPU consumidas por su carga de trabajo.

Horas de nodo

Para crear un clúster basado en nodos, elija la familia de nodos de EC2, el tamaño, la cantidad de nodos y la ubicación en las zonas de disponibilidad. Al crear un clúster basado en nodos, se paga por hora de uso de cada nodo de caché.

Copias de seguridad de ElastiCache

Una copia de seguridad es una copia en un momento dado de una caché sin servidor o de un clúster basado en nodos de Valkey o Redis OSS. ElastiCache le permite realizar una copia de seguridad de sus datos en cualquier momento o configurar copias de seguridad automáticas. Dichas copias de seguridad pueden usarse para restaurar una caché existente o para propagar datos en una nueva caché. Las copias de seguridad se componen de todos los datos en una caché, además de algunos metadatos. Para obtener más información consulte () Instantánea y restauración.

Cómo elegir una opción de implementación

Amazon ElastiCache tiene dos opciones de implementación:

- Almacenamiento en caché sin servidor
- Clústeres basados en nodos

Para obtener una lista de comandos admitidos para ambos, consulte Comandos de Valkey, Memcached y Redis OSS compatibles y restringidos.

Almacenamiento en caché sin servidor

Amazon ElastiCache Serverless simplifica la creación de memoria caché y se escala al instante para dar soporte a las aplicaciones más exigentes de los clientes. Con ElastiCache sin servidor, puede crear una caché escalable y de alta disponibilidad en menos de un minuto, lo que elimina la necesidad de aprovisionar, planificar y administrar la capacidad del clúster. ElastiCache Serverless almacena automáticamente los datos de forma redundante en tres zonas de disponibilidad y proporciona un acuerdo de nivel de servicio (SLA) con una disponibilidad del 99,99 %. Las copias de seguridad de los clústeres basados en nodos de Valkey o Redis OSS se pueden restaurar en una configuración sin servidor.

Clústeres basados en nodos

Si necesita un control detallado del clúster de Valkey, Memcached o Redis OSS, puede crear un clúster basado en nodos con ElastiCache. Puede elegir el tipo de nodo, la cantidad de nodos y la ubicación de los nodos en las zonas de disponibilidad de AWS para el clúster. Como ElastiCache es un servicio totalmente administrado, ayuda a administrar el aprovisionamiento de hardware, la supervisión, los reemplazos de nodos y la aplicación de parches de software para el clúster. Los clústeres basados en nodos se pueden diseñar para ofrecer un SLA de disponibilidad de hasta el

99,99 %. Las copias de seguridad de las cachés de Valkey o Redis OSS sin servidor se pueden restaurar en un clúster basado en nodos.

Cómo elegir una opción de implementación

Elija el almacenamiento en caché sin servidor en los siguientes casos:

- Va a crear una caché para cargas de trabajo nuevas o difíciles de predecir.
- El tráfico de la aplicación es impredecible.
- Desea optar por la forma más sencilla de comenzar a utilizar una caché.

Cree su propio clúster basado en nodos si:

- Ya utiliza ElastiCache sin servidor y desea tener un control más preciso sobre el tipo de nodo que ejecuta Valkey, Memcached o Redis OSS, el número de nodos y la ubicación de los mismos.
- Espera que el tráfico de sus aplicaciones sea relativamente predecible y desea tener un control detallado del rendimiento, la disponibilidad y el costo.
- Puede prever los requisitos de capacidad para controlar los costes.

Comparación entre el almacenamiento en caché sin servidor y los clústeres basados en nodos

Característica	Almacenamiento en caché sin servidor	Clústeres basados en nodos
Configuración de caché	Cree una caché únicamente con un nombre en menos de un minuto	Proporciona un control detallado sobre el diseño del clúster. El usuario puede elegir el tipo de nodo, la cantidad de nodos y la ubicación en las distintas zonas de disponibilidad de AWS
Versión de ElastiCache compatible	Valkey 7.2 y versiones posteriores, Redis OSS versión 7.1 y posterior	Valkey 7.2 y versiones posteriores, Redis OSS versión 4.0 y posteriores,

Característica	Almacenamiento en caché sin servidor	Clústeres basados en nodos
	es, Memcached 1.6.21 y versiones posteriores	Memcached 1.4 y versiones posteriores
Modo de clúster (Valkey y Redis OSS)	Hace funcionar los motores únicamente en `cluster mode enabled`. Los clientes deben admitir `cluster mode enabled` para conectarse a ElastiCache sin servidor.	Se puede configurar para que funcione en el modo de clúster habilitado o en el modo de clúster deshabilitado.

Característica	Almacenamiento en caché sin servidor	Clústeres basados en nodos
Escalado	Escala automáticamente los motores tanto vertical como horizontalmente sin administrar la capacidad.	Permite controlar el escalado y, al mismo tiempo, requiere supervisión para garantizar que la capacidad actual satisfaga adecuadamente la demanda. Para Valkey y Redis OSS, puede escalar verticalmente aumentando o disminuyendo el tamaño del nodo de caché cuando sea necesario. Además, puede escalar horizontalmente añadiendo particiones nuevas o añadiendo más réplicas a sus particiones. Esta capacidad no está disponible para Memcached. Con la característica de escalado automático, también puede configurar el escalado en función de un cronograma o escalar en función de métricas como el uso de CPU y memoria en la caché.

Característica	Almacenamiento en caché sin servidor	Clústeres basados en nodos
Conexiones de clientes	Los clientes se conectan a un único punto de conexión. Esto permite que la topología del nodo de caché subyacente (escalado, reemplazos y actualizaciones) cambie sin desconectar el cliente.	Los clientes se conectan a cada nodo de caché individual. Si se reemplaza un nodo, el cliente vuelve a descubrir la topología del clúster y restablece las conexiones.
Capacidad de configuración	No hay disponible una configuración detallada. Los clientes pueden configurar los ajustes básicos, como las subredes que pueden acceder a la caché, si se activan o desactivan las copias de seguridad automáticas y los límites máximos de uso de la caché.	Los clústeres basados en nodos ofrecen opciones de configuración detalladas. Los clientes pueden usar grupos de parámetros para disfrutar de un control detallado. Para ver una tabla de estos valores de los parámetros por tipo de nodo, consulte Parámetros específicos del motor.
Multi-AZ	Los datos se replican de forma asíncrona en varias zonas de disponibilidad para aumentar la disponibilidad y mejorar la latencia de lectura.	Ofrece una opción para crear el clúster en una única zona de disponibilidad o en varias zonas de disponibilidad (AZ). Al utilizar Valkey o Redis OSS, proporciona clústeres multi-AZ con datos replicados de forma asíncrona en varias zonas de disponibilidad para aumentar la disponibilidad y mejorar la latencia de lectura.

Característica	Almacenamiento en caché sin servidor	Clústeres basados en nodos
Cifrado en reposo	Siempre habilitado. Los clientes pueden utilizar una Clave administrada de AWS o una clave administrada por el cliente en AWS KMS.	Opción de habilitar o deshabilitar el cifrado en reposo. Cuando está habilitada, los clientes pueden utilizar una Clave administrada de AWS o una clave administrada por el cliente en AWS KMS.
Cifrado en tránsito (TLS)	Siempre habilitado. Los clientes deben admitir la conectividad de TLS.	Opción para habilitar o deshabilitar.
Copias de seguridad	Admite copias de seguridad automáticas y manuales de las cachés sin que ello afecte al rendimiento. Las copias de seguridad de Valkey y Redis OSS son compatibles entre sí y se pueden restaurar en una caché de ElastiCache sin servidor o en un clúster basado en nodos.	Admite copias de seguridad automáticas y manuales para Valkey y Redis OSS. El rendimiento de los clústeres puede verse afectado en cierta medida en función de la memoria reservada disponible. Para obtener más información, consulte Administración de la memoria reservada para Valkey y Redis OSS. Las copias de seguridad de Valkey y Redis OSS son compatibles entre sí y se pueden restaurar en una caché de ElastiCache sin servidor o en un clúster basado en nodos.

Característica	Almacenamiento en caché sin servidor	Clústeres basados en nodos
Monitorización	Admite métricas de caché, incluida la tasa de aciertos de caché, la tasa de errores de caché, el tamaño de los datos y las ECPU consumidas. ElastiCache sin servidor envía eventos mediante EventBridge cuando se producen eventos importantes en la caché. Puede elegir monitorizar, ingerir, transformar y actuar en función de los eventos de ElastiCache mediante Amazon EventBridge. Para obtener más información, consulte Eventos de caché sin servidor.	Los clústeres basados en nodos de ElastiCache emiten métricas en cada nivel de nodo, incluidas métricas de host y métricas de caché. Los clústeres basados en nodos emiten notificaciones de SNS en caso de eventos importantes. Consulte Métricas de Memcached y Métricas de Valkey y Redis OSS.
Disponibilidad	Acuerdo de nivel de servicio (SLA) con una disponibilidad del 99,99 %	Los clústeres basados en nodos se pueden diseñar para alcanzar un acuerdo de nivel de servicio (SLA) de disponibilidad de hasta el 99,99 %, según la configuración.

Característica	Almacenamiento en caché sin servidor	Clústeres basados en nodos
Actualizaciones de software y parches	Actualiza automáticamente el software de caché a la versión secundaria y la versión de parche más recientes, sin que ello afecte a la aplicación. Los clientes reciben una notificac ión sobre las actualizaciones de las versiones principales y pueden actualizar a la versión principal más reciente cuando lo deseen.	Los clústeres basados en nodos ofrecen una función de autoservicio habilitada por el cliente para realizar las actualizaciones de versiones secundarias y parches, así como para actualizaciones de versiones principales. Las actualizaciones administr adas se aplican automátic amente durante el período de mantenimiento definido por el cliente. Los clientes también pueden optar por aplicar una actualización de la versión secundaria o de un parche bajo pedido.
Almacén de datos global	No admitido	Admite el almacén de datos global, que permite la replicaci ón entre regiones con escrituras en una sola región y lecturas en varias regiones

Característica	Almacenamiento en caché sin servidor	Clústeres basados en nodos
Organización de datos en niveles	No admitido	Los clústeres que se han creado utilizando nodos de la familia r6gd tienen sus datos en niveles entre la memoria y el almacenamiento de SSD local (unidades de estado sólido). La organización de datos en niveles proporciona una opción con una buena relación precio-rendimiento para las cargas de trabajo de Valkey y Redis OSS mediante el uso de unidades de estado sólido (SSD) de menor costo en cada nodo de clúster, además de almacenar datos en la memoria.
Modelo de precios	Pago por uso, en función de los datos almacenados en GB/hora y las solicitudes en unidades de procesamiento de ElastiCache (o ECPU). Puede encontrar más información sobre los precios aquí.	Pago por hora, en función del uso de los nodos de caché. Puede encontrar más información sobre los precios aquí.

Temas relacionados:

- Creación y administración de un clúster de ElastiCache basado en nodos

Recursos de Amazon ElastiCache para usuarios nuevos

Recomendamos a los nuevos usuarios que empiecen por leer las secciones siguientes y que las consulten cuando sea necesario.

- Precios y aspectos resaltados del servicio: en la página de detalles del producto se proporciona información general sobre ElastiCache junto con los aspectos resaltados del servicio y los precios.
- Videos de ElastiCache: la sección Videos de ElastiCache contiene videos que presentan a Amazon ElastiCache. En los videos se cubren casos de uso frecuentes de ElastiCache y se muestran cómo utilizar ElastiCache para reducir la latencia y mejorar el rendimiento de las aplicaciones.
- Introducción: la sección Introducción a Amazon ElastiCache incluye información sobre la creación de una caché. Explica también cómo autorizar el acceso al clúster de caché, conectarse a un nodo de caché y eliminar el caché.
- Rendimiento a escala: en el documento técnico Performance at scale with Amazon ElastiCache se abordan estrategias de almacenamiento en caché que asisten a la aplicación para realizar un escalado correcto.

Cuando complete las secciones anteriores, lea estas secciones:

- Elección del tamaño del nodo

  Desea que los nodos sean lo suficientemente grandes como para alojar todos los datos que necesite almacenar en caché. Al mismo tiempo, no desea pagar más caché de la que necesita. Puede utilizar este tema para ayudarlo a seleccionar el mejor tamaño de nodo.
- Procedimientos recomendados y estrategias de almacenamiento en caché de ElastiCache

  Identifique y aborde los problemas que pueden influir en la eficacia del clúster.

Si desea utilizar la AWS Command Line Interface (AWS CLI), puede utilizar estos documentos, que lo ayudarán a empezar:

- AWS Command Line Interface Documentación de

  En esta sección, se proporciona información sobre cómo descargar la AWS CLI, poner a funcionar la AWS CLI en su sistema y proporcionar sus credenciales de AWS.
- AWS CLI Documentación de la para ElastiCache

Este documento independiente abarca todas las AWS CLI para los comandos de ElastiCache, incluida la sintaxis, además de ejemplos.

Puede escribir programas de aplicación para utilizar la API de ElastiCache con diversos lenguajes de programación conocidos. Estos son algunos de los recursos:

- Herramientas para Amazon Web Services

  Amazon Web Services proporciona una serie de kits de desarrollo de software (SDK) compatibles con ElastiCache. Puede codificar para ElastiCache con Java, .NET, PHP, Ruby y otros lenguajes. Estos SDK pueden simplificar en gran medida el desarrollo de la aplicación al aplicar formato a sus solicitudes a ElastiCache, analizar las respuestas y proporcionar gestión de errores y lógica de reintentos.
- Uso de la API de ElastiCache

  Si no desea utilizar los SDK de AWS, puede interactuar con ElastiCache directamente a través de la API de consulta. En esta sección, puede encontrar consejos de solución de problemas e información acerca de la creación y autenticación de solicitudes, así como de la gestión de respuestas.
- Referencia de la API de Amazon ElastiCache

  Este documento independiente abarca todas las operaciones de la API de ElastiCache, incluida la sintaxis, además de ejemplos.

AWSRegiones y zonas de disponibilidad de

Los recursos de informática en la nube de Amazon están alojados en instalaciones de centros de datos con alta disponibilidad, en diferentes zonas del mundo (por ejemplo, Norteamérica, Europa o Asia). Cada ubicación del centro de datos se denomina AWS región.

Cada AWS región contiene varias ubicaciones distintas denominadas zonas de disponibilidad o AZ. Cada zona de disponibilidad está diseñada para quedar aislada en caso de error en otras zonas de disponibilidad. Cada una de ellas está diseñada para proporcionar conectividad de red económica y de baja latencia a otras zonas de disponibilidad de la misma AWS región. Al lanzar instancias en distintas zonas de disponibilidad, puede proteger sus aplicaciones de los errores que se produzcan en una única ubicación. Para obtener más información, consulte Elección de regiones y zonas de disponibilidad.

Puede crear un clúster en varias zonas de disponibilidad, lo que recibe el nombre de implementación Multi-AZ. Cuando elige esta opción, Amazon aprovisiona y mantiene una instancia de nodos secundaria en espera en una zona de disponibilidad diferente de forma automática. Su instancia de nodo primario se replica de forma asincrónica en las zonas de disponibilidad para la instancia secundaria. Este enfoque ayuda a proporcionar redundancia de datos y soporte de conmutación por error, elimina los bloqueos de E/S y minimiza los picos de latencia durante las copias de seguridad del sistema. Para obtener más información, consulte Minimización del tiempo de inactividad en ElastiCache para Valkey y Redis OSS con Multi-AZ.

Casos de uso de ElastiCache comunes y cómo puede ayudar ElastiCache

Tanto si se trata de la publicación de las últimas noticas, de una tabla de clasificación Top 10, de un catálogo de productos o de la venta de entradas para un espectáculo, la velocidad es un factor clave a tener en cuenta. El éxito de su sitio web y de su negocio depende considerablemente de la rapidez con la que se entrega el contenido.

En el artículo del New York Times "For Impatient Web Users, an Eye Blink Is Just Too Long to Wait" (Para un usuario impaciente de Internet, un abrir y cerrar de ojos ya es esperar demasiado), se observó que los usuarios pueden percibir una diferencia de 250 milisegundos (1/4 de segundo) entre los sitios competidores. Los usuarios suelen descartar el sitio más lento en favor del más rápido. Las pruebas realizadas en Amazon, citadas en How Webpage Load Time Is Related to Visitor Loss (Relación entre el tiempo de carga de la página web y la pérdida de visitantes), revelaron que por cada incremento de 100 ms (1/10 de segundo) de tiempo de carga, las ventas disminuían un 1 %.

Si alguien necesita tener acceso a estos datos, puede entregar esos datos mucho más rápido si se almacenan en caché. Eso es cierto, independientemente de si se trata de una página web o de un informe que impulsa las decisiones empresariales. ¿Puede su negocio permitirse no almacenar en caché sus páginas web con el fin de ofrecerlas con la menor latencia posible?

Puede parecer intuitivamente evidente que desee almacenar en caché los elementos más solicitados. Pero ¿por qué no en almacenar en la caché los elementos solicitados con menos frecuencia? Incluso la consulta de la base de datos más optimizada o las llamadas a la API remota van a ser notablemente más lentas que recuperar una clave plana en una caché en memoria. El concepto de notablemente más lento suele mandar a los clientes a otro sitio.

En los siguientes ejemplos, se ilustran algunas de las formas en las que el uso de ElastiCache puede mejorar el rendimiento general de su aplicación.

Temas

- Almacenamiento de datos en memoria
- Tablas de clasificación de juegos
- Mensajería (publicación/suscripción)
- Datos de recomendación (valores hash)
- Almacenamiento en caché semántico para aplicaciones de IA generativa
- Testimonios de los clientes de ElastiCache

Almacenamiento de datos en memoria

El propósito principal del almacenamiento en memoria de pares clave-valor es proporcionar un acceso ultrarrápido (latencia de milisegundos) y económico a las copias de datos. La mayoría de almacenes de datos tienen áreas de datos a los que se accede con frecuencia, pero que raramente se actualizan. Además, la consulta de una base de datos siempre será más lenta y más cara que la localización de una clave en una caché de par clave-valor. Algunas consultas de base de datos son especialmente costosas de realizar. Por ejemplo, las consultas que implican uniones entre varias tablas o las consultas con cálculos intensivos. Al almacenar en caché resultados de una consulta de este tipo, solo pagará el precio de la consulta una vez. A continuación, puede recuperar rápidamente los datos varias veces sin tener que volver a ejecutar la consulta.

¿Qué debo almacenar en caché?

A la hora de decidir qué datos almacenar en caché, tenga en cuenta estos factores:

Velocidad y gastos: siempre es más lento y más caro adquirir los datos de una base de datos que de una caché. Algunas consultas de base de datos son intrínsecamente más lentas y más caras que otras. Por ejemplo, las consultas que realizan uniones en varias tablas son mucho más lentas y más caras que las consultas sencillas en una única tabla. Si los datos interesantes que se desean adquirir requieren una consulta lenta y cara, son un candidato para el almacenamiento en caché. Si la adquisición de los datos requiere una consulta relativamente rápida y sencilla, puede ser un candidato para el almacenamiento en caché, en función de otros factores.

Patrón de datos y acceso: determinar qué almacenar en caché también implica comprender los datos en sí y sus patrones de acceso. Por ejemplo, no tiene sentido almacenar en caché los datos a los que se accede raramente o que cambian rápidamente. Para que el almacenamiento en caché proporcione un beneficio real, los datos deben ser relativamente estáticos y el acceso a estos debería ser frecuente. Un ejemplo es un perfil personal en un sitio de redes sociales. Por otro lado, no deseará almacenar en caché datos si dicho almacenamiento no supone ventajas de velocidad o costos. Por ejemplo, no tiene sentido almacenar en caché páginas web que devuelven los resultados de búsqueda, ya que estas consultas y sus resultados suelen ser únicos.

Estancamiento: por definición, los datos almacenados en caché son datos obsoletos. Incluso si en ciertas circunstancias no se encuentran obsoletos, siempre deben considerarse y tratarse como obsoletos. Para determinar si sus datos son un candidato para el almacenamiento en caché, debe evaluar la tolerancia de su aplicación a los datos obsoletos.

Es posible que la aplicación sea capaz de tolerar datos obsoletos en un contexto, pero no en otro. Por ejemplo, suponga que su sitio web sirve a un precio bursátil. Sus clientes podrían aceptar cierto

estancamiento con una exención de responsabilidad de que los precios podrían tener un retraso de n minutos. Sin embargo, si presenta ese precio a un agente que realiza una venta o una compra, deseará tener datos en tiempo real.

Considere la posibilidad de almacenar en caché los datos si:

- Resultan lentos o caros de adquirir en comparación con la recuperación en caché.
- Los usuarios acceden a sus datos con frecuencia.
- Sus datos se mantienen relativamente iguales, o si cambian con rapidez, el estancamiento no es un problema importante.

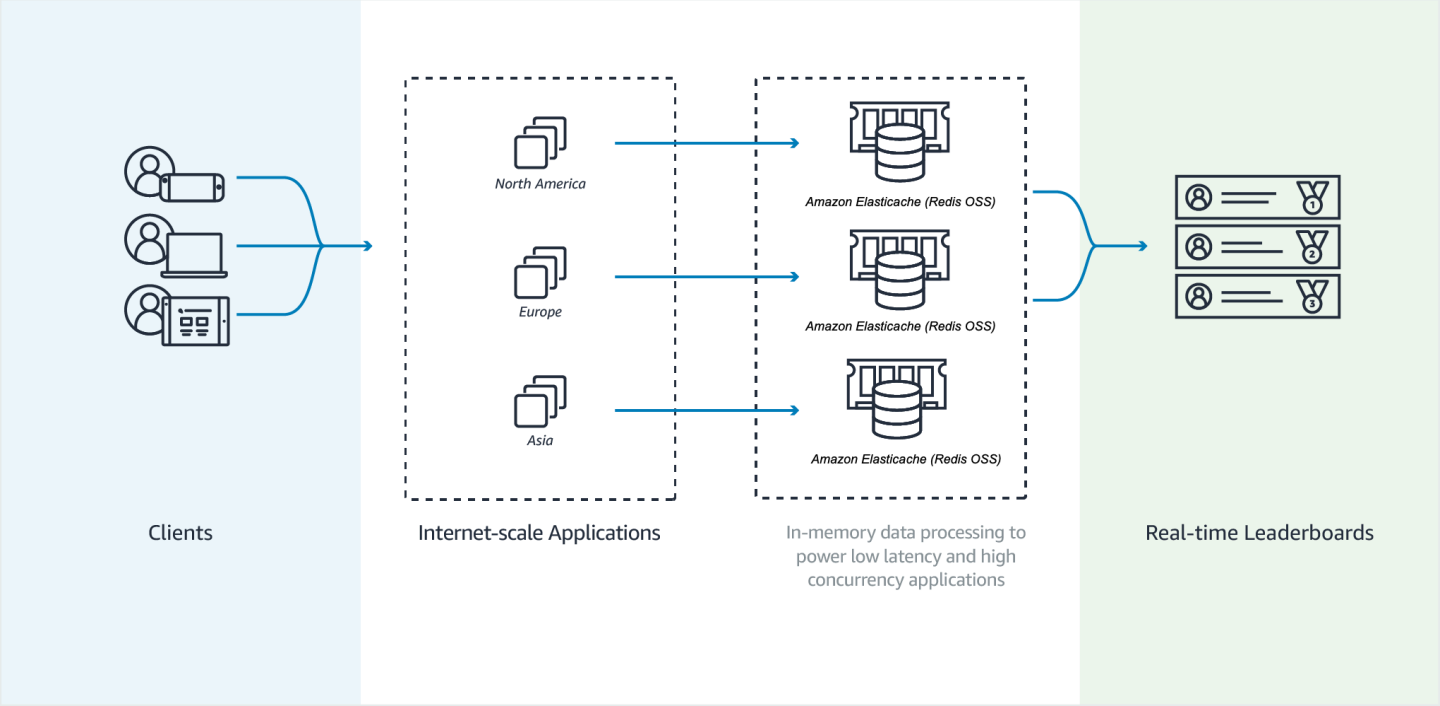
Para obtener más información, consulte Estrategias de almacenamiento en caché para Memcached

Tablas de clasificación de juegos

Con los conjuntos ordenados de Valkey o Redis OSS, puede trasladar la complejidad computacional de las tablas de clasificación de su aplicación a su clúster.

Las tablas de clasificación, como las de 10 mejores puntuaciones de un juego, son complejas en términos informáticos. Esto es especialmente cierto cuando hay un gran número de jugadores simultáneos y las puntuaciones cambian de forma constante. Los conjuntos ordenados de Valkey y Redis OSS garantizan tanto la exclusividad como la ordenación de los elementos. Con los conjuntos ordenados, cada vez que se añade un nuevo elemento al conjunto ordenado, este se vuelve a clasificar en tiempo real. A continuación, se agrega al conjunto en el orden numérico correcto.

En el siguiente diagrama, puede ver cómo funciona una tabla de clasificación de juegos de ElastiCache.



Example Tabla de clasificación de Valkey o Redis OSS

En este ejemplo, cuatro jugadores y sus puntuaciones se introducen en una lista ordenada mediante ZADD. El comando ZREVRANGEBYSCORE muestra a los jugadores por su puntuación, de más alta a más baja. A continuación, ZADD se utiliza para actualizar la puntuación de June, sobrescribiendo la entrada existente. Por último, ZREVRANGEBYSCORE muestra a los jugadores por su puntuación, de más alta a más baja. La lista muestra que June ha ascendido en la clasificación.

```
ZADD leaderboard 132 Robert
ZADD leaderboard 231 Sandra
ZADD leaderboard 32 June
ZADD leaderboard 381 Adam

ZREVRANGEBYSCORE leaderboard +inf -inf
1) Adam
2) Sandra
3) Robert
4) June

ZADD leaderboard 232 June

ZREVRANGEBYSCORE leaderboard +inf -inf
1) Adam
2) June
```

```
3) Sandra
4) Robert
```

El siguiente comando permite a June conocer dónde se sitúa entre todos los jugadores. Dado que la clasificación está basada en cero, ZREVRANK devuelve un 1 para June, que está en segunda posición.

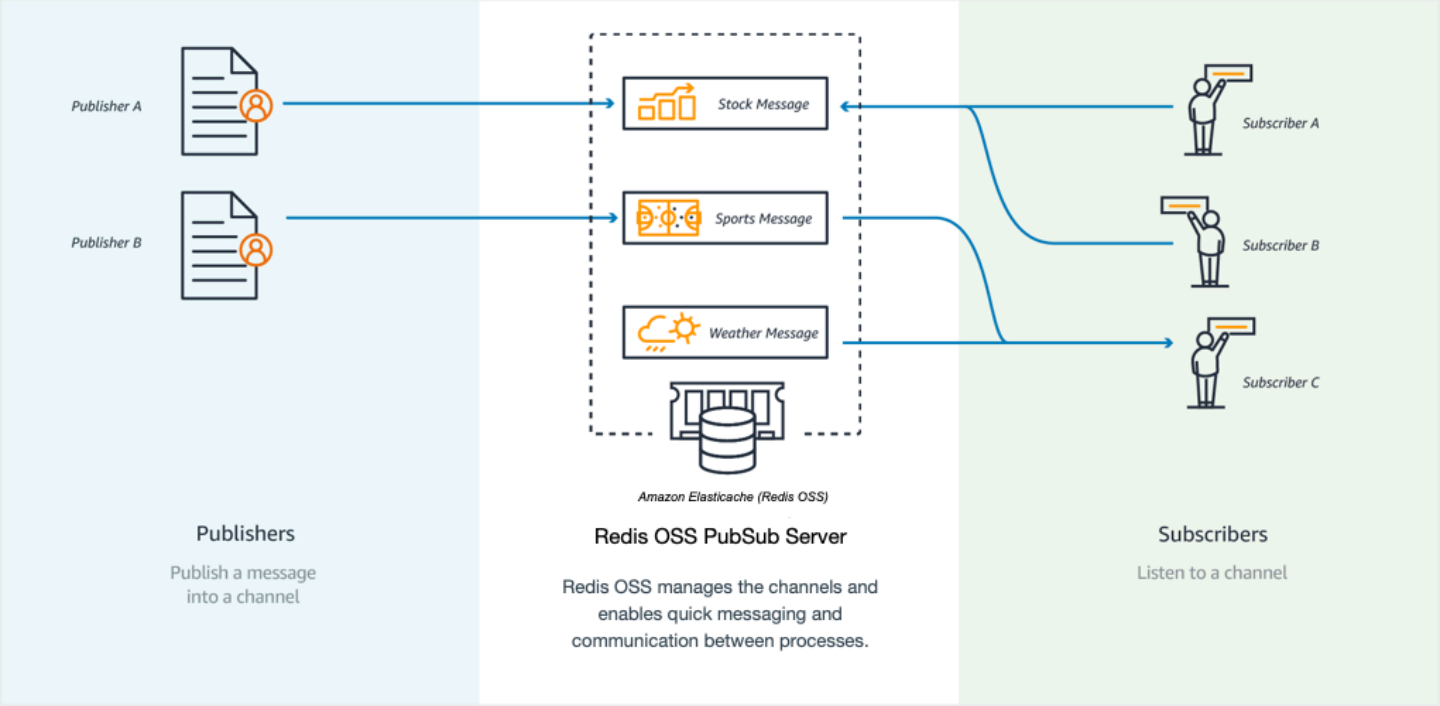
```
ZREVRANK leaderboard June
1
```

Para obtener más información, consulte la documentación de Valkey acerca de los conjuntos ordenados.

Mensajería (publicación/suscripción)

Al enviar un mensaje de correo electrónico, lo envía a uno o a varios destinatarios especificados. En el paradigma de publicación/suscripción de Valkey y Redis OSS, se envía un mensaje a un canal específico sin saber quién lo recibirá y si lo recibirá. Las personas que reciben el mensaje son aquellas que se encuentran suscritas al canal. Por ejemplo, suponga que se suscribe al canal news.sports.golf. Usted y todos los demás que se hayan suscrito al canal news.sports.golf recibirán todos los mensajes que se publiquen en news.sports.golf.

La funcionalidad de publicación/suscripción de Redis no tiene ninguna relación con ningún espacio de claves. Por lo tanto, no interferirá en ningún nivel. En el siguiente diagrama, puede ver una ilustración de la mensajería de ElastiCache con Valkey y Redis OSS.



Suscripción

Para recibir mensajes en un canal, debe suscribirse a dicho canal. Puede suscribirse a un solo canal, a varios canales especificados o a todos los canales que coincidan con un patrón. Para cancelar una suscripción, debe cancelar la suscripción al canal especificado en el momento de la suscripción. O bien, si se ha suscrito mediante la coincidencia de patrones, anula la suscripción con el mismo patrón que utilizó antes.

Example - Suscripción a un único canal

Para suscribirse a un solo canal, utilice el comando SUBSCRIBE y especifique el canal al que se desea suscribir. En el siguiente ejemplo, un cliente se suscribe al canal news.sports.golf.

```
SUBSCRIBE news.sports.golf
```

Después de un tiempo, el cliente cancela su suscripción al canal con el comando UNSUBSCRIBE y especifica el canal del que desea cancelar la suscripción.

```
UNSUBSCRIBE news.sports.golf
```

Example - Suscripciones a varios canales especificados

Para suscribirse a varios canales específicos, enumere los canales con el comando SUBSCRIBE. En el siguiente ejemplo, un cliente se suscribe a los canales news.sports.golf, news.sports.soccer y news.sports.skiing

```
SUBSCRIBE news.sports.golf news.sports.soccer news.sports.skiing
```

Para cancelar una suscripción a un canal específico, utilice el comando UNSUBSCRIBE y especifique el canal del que desea cancelar la suscripción.

```
UNSUBSCRIBE news.sports.golf
```

Para cancelar suscripciones a varios canales, utilice el comando UNSUBSCRIBE y especifique los canales de los que desea cancelar la suscripción.

```
UNSUBSCRIBE news.sports.golf news.sports.soccer
```

Para cancelar todas las suscripciones, utilice `UNSUBSCRIBE` y especifique cada canal. O utilice `UNSUBSCRIBE` y no especifique un canal.

```
UNSUBSCRIBE news.sports.golf news.sports.soccer news.sports.skiing
```

o

```
UNSUBSCRIBE
```

Example - Suscripciones mediante la coincidencia de patrones

Los clientes pueden suscribirse a todos los canales que coincidan con un patrón con el comando PSUBSCRIBE.

En el siguiente ejemplo, un cliente se suscribe a todos los canales de deportes. No enumera todos los canales de deportes de manera individual, como se haría con `SUBSCRIBE`. En su lugar, con el comando `PSUBSCRIBE`, se utiliza la coincidencia de patrones.

```
PSUBSCRIBE news.sports.*
```

Example Cancelación de suscripciones

Para cancelar las suscripciones a estos canales, utilice el comando PUNSUBSCRIBE.

```
PUNSUBSCRIBE news.sports.*
```

> **Important**
>
> - La cadena de canales enviada al comando [P]SUBSCRIBE y la enviada al comando [P]UNSUBSCRIBE deben coincidir. No puede PSUBSCRIBE a news.* y PUNSUBSCRIBE de news.sports.* o UNSUBSCRIBE de news.sports.golf.
> - PSUBSCRIBE y PUNSUBSCRIBE no están disponibles para ElastiCache sin servidor.

Publicación

Para enviar un mensaje a todos los suscriptores de un canal, utilice el comando PUBLISH y especifique el canal y el mensaje. En el siguiente ejemplo, se publica el mensaje: “It’s Saturday and sunny. I’m headed to the links” (Es sábado y está soleado. Vayamos directamente a los enlaces). en el canal news.sports.golf.

```
PUBLISH news.sports.golf "It's Saturday and sunny. I'm headed to the links."
```

Un cliente no puede publicar en un canal al que está suscrito.

Para obtener más información, consulte el tema sobre la publicación/suscripción en la documentación de Valkey.

Datos de recomendación (valores hash)

El uso de INCR o DECR en Valkey o Redis OSS simplifica la compilación de recomendaciones. Cada vez que un usuario da un “me gusta” a un producto, aumenta un contador item:productID:like. Cada vez que un usuario da un “no me gusta” a un producto, aumenta un contador item:productID:dislike. Al utilizar hash, también puede mantener una lista de todos los que han indicado que les gusta o no les gusta un producto.

Example - Me gusta y no me gusta

```
INCR item:38923:likes
```

```
HSET item:38923:ratings Susan 1
INCR item:38923:dislikes
HSET item:38923:ratings Tommy -1
```

Almacenamiento en caché semántico para aplicaciones de IA generativa

Operar aplicaciones de IA generativa a gran escala puede resultar difícil debido al costo y la latencia asociados a las llamadas de inferencia a modelos de lenguaje de gran tamaño (LLM). Puede usar ElastiCache para el almacenamiento en caché semántico en aplicaciones de IA generativa, lo que le permite reducir el costo y la latencia de las llamadas de inferencia a LLM. Con el almacenamiento en caché semántico, puede devolver una respuesta almacenada en caché mediante una coincidencia basada en vectores para encontrar similitudes entre las peticiones actuales y las anteriores. Si la petición de un usuario es semánticamente similar a una anterior, se devolverá una respuesta en caché en lugar de realizar una nueva llamada de inferencia a LLM, lo que reduce el costo de las aplicaciones de IA generativa y proporciona respuestas más rápidas que mejoran la experiencia del usuario. Puede controlar qué consultas se envían a la memoria caché configurando umbrales de similitud para las peticiones y aplicando filtros de metadatos numéricos o de etiquetas.

Las actualizaciones de índices en línea en tiempo real que proporciona la búsqueda vectorial para ElastiCache ayudan a garantizar que la memoria caché se actualice de forma continua a medida que llegan las peticiones de los usuarios y las respuestas de los LLM. Esta indexación en tiempo real es fundamental para mantener actualizados los resultados almacenados en caché y las tasas de aciertos de caché, especialmente en el caso de los picos de tráfico. Además, ElastiCache simplifica las operaciones de almacenamiento en caché semántico mediante primitivas de caché maduras, como los TTL por clave, estrategias de expulsión configurables, operaciones atómicas y una amplia estructura de datos y compatibilidad con scripts.

Memoria para agentes y asistentes de IA generativa

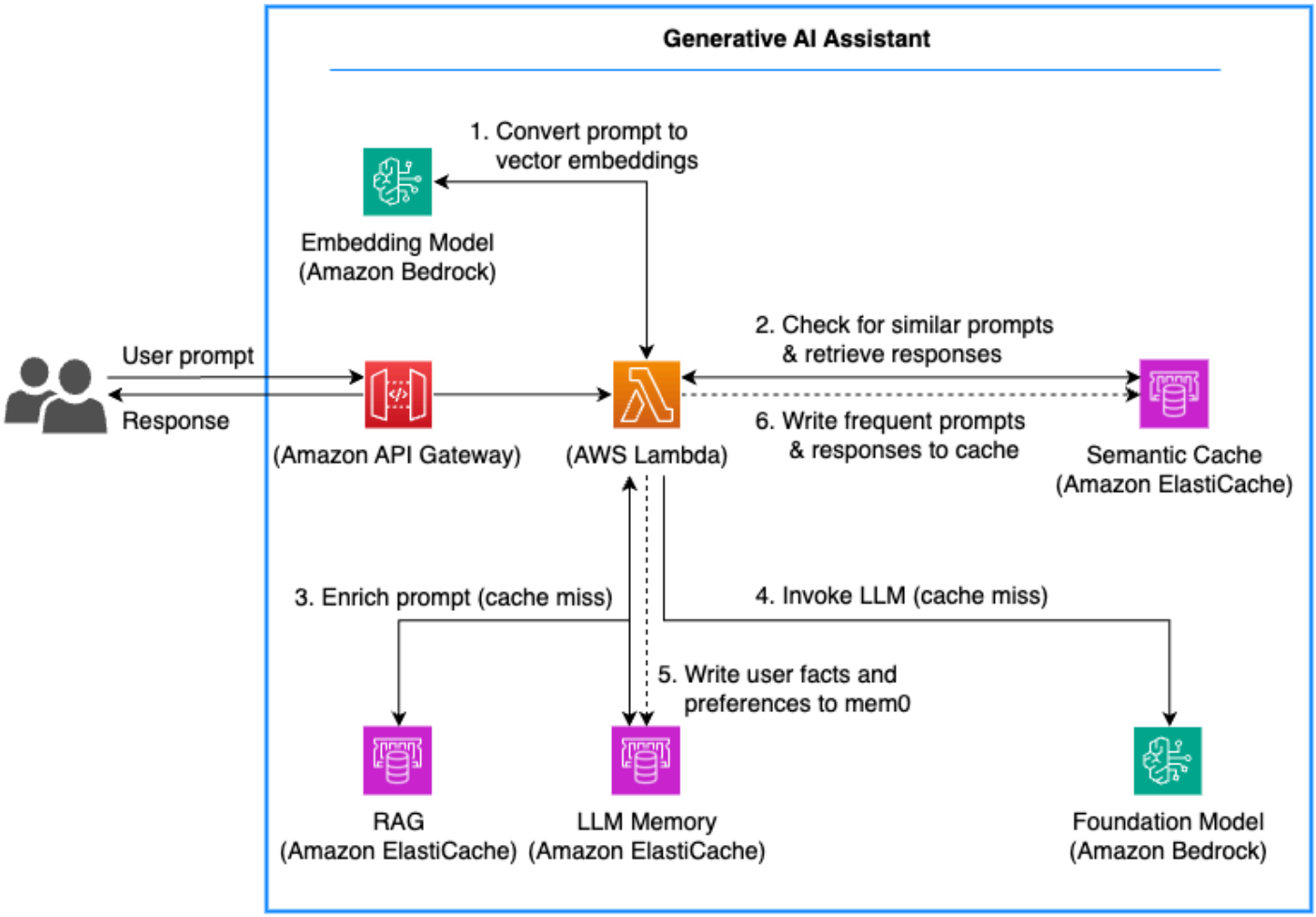
Puede usar ElastiCache para ofrecer respuestas más personalizadas y sensibles al contexto mediante la implementación de mecanismos de memoria que muestran el historial de conversaciones entre sesiones a los LLM. La memoria conversacional permite a los asistentes y agentes de IA generativa retener y utilizar las interacciones pasadas para personalizar las respuestas y mejorar la relevancia. Sin embargo, el simple hecho de agregar todas las interacciones anteriores en la petición no es efectivo, ya que los tokens adicionales irrelevantes aumentan el costo, degradan la calidad de la respuesta y corren el riesgo de superar el contexto del LLM. En su lugar, puede utilizar la búsqueda vectorial para recuperar y proporcionar solo los datos pertinentes en el contexto de cada invocación del LLM.

ElastiCache para Valkey ofrece integraciones con capas de memoria de código abierto, lo que proporciona conectores integrados para almacenar y recuperar memorias para las aplicaciones y los agentes de LLM. La búsqueda vectorial para ElastiCache proporciona actualizaciones rápidas de los índices, lo que mantiene la memoria actualizada y permite buscar nuevos recuerdos de forma inmediata. La búsqueda vectorial de baja latencia agiliza las búsquedas en memoria, lo que permite implementarlas en la ruta en línea de cada solicitud, no solo en tareas en segundo plano. Además de la búsqueda vectorial, ElastiCache para Valkey también proporciona primitivas de almacenamiento en caché para el estado de la sesión, las preferencias del usuario y los indicadores de características, lo que proporciona un único servicio para almacenar el estado de la sesión de corta duración y las memorias a largo plazo en un almacén de datos.

Generación aumentada por recuperación (RAG)

La RAG es el proceso que consiste en proporcionar a los LLM información actualizada en la petición para mejorar la relevancia de las respuestas. La RAG reduce las alucinaciones y mejora la precisión de los hechos al basar los resultados en orígenes de datos del mundo real. Las aplicaciones RAG utilizan la búsqueda vectorial para obtener contenido semánticamente relevante de una base de conocimientos. La búsqueda vectorial de baja latencia proporcionada por ElastiCache es adecuada para implementar RAG en cargas de trabajo que tienen conjuntos de datos grandes con millones o más vectores. Además, la compatibilidad con las actualizaciones de índices vectoriales en línea hace que ElastiCache sea adecuado para los asistentes con flujos de trabajo de carga que necesitan garantizar que cualquier dato cargado se pueda buscar inmediatamente. La tecnología RAG en los sistemas de IA agéntica garantiza que los agentes dispongan de información actualizada para poder realizar acciones precisas. La búsqueda vectorial de baja latencia también es crucial para la RAG en los sistemas de IA agéntica, ya que una sola consulta puede desencadenar varias llamadas a LLM y acumular la latencia de la búsqueda vectorial subyacente.

En el siguiente diagrama se muestra un ejemplo de arquitectura que utiliza ElastiCache para implementar una caché semántica, mecanismos de memoria y una RAG para mejorar una aplicación de IA generativa en producción.



Búsqueda semántica

La búsqueda vectorial recupera los datos de texto, voz, imagen o vídeo más relevantes en función de la proximidad con el significado o las características. Esta capacidad permite que las aplicaciones de machine learning se basen en la búsqueda por similitud en diversas modalidades de datos, incluidos los motores de recomendación, la detección de anomalías, la personalización y los sistemas de gestión del conocimiento. Los sistemas de recomendación utilizan representaciones vectoriales para capturar patrones complejos en el comportamiento de los usuarios y las características de los elementos, lo que les permite sugerir el contenido más pertinente. La búsqueda vectorial para ElastiCache es adecuada para estas aplicaciones debido a las actualizaciones prácticamente en tiempo real y a su baja latencia, lo que permite realizar comparaciones de similitudes que ofrecen recomendaciones instantáneas y muy relevantes basadas en las interacciones de los usuarios en tiempo real.

Testimonios de los clientes de ElastiCache

Para obtener información sobre cómo empresas como Airbnb, PBS, Esri y otros utilizan Amazon ElastiCache a fin de impulsar sus negocios con una mejor experiencia del cliente, consulte Cómo utilizan Amazon ElastiCache otros clientes.

También puede ver los tutoriales de vídeo para conocer más casos de uso de clientes de ElastiCache.

Introducción a Amazon ElastiCache

Utilice el tutorial práctico de esta sección como ayuda para empezar y aprender más sobre el uso de ElastiCache.

Temas

- Configuración de ElastiCache
- Creación de una caché sin servidor de Valkey
- Creación de una caché sin servidor de Redis OSS
- Creación de una caché sin servidor de Memcached
- Tutoriales: introducción a Python y ElastiCache
- Tutorial: configuración de Lambda para acceder a ElastiCache en una VPC

Configuración de ElastiCache

Para usar el servicio web de ElastiCache, siga estos pasos.

Temas

- Cómo crear una Cuenta de AWS
- Creación de un usuario con acceso administrativo
- Conceder acceso programático
- Configurar los permisos (solo para nuevos usuarios de ElastiCache)
- Configuración de EC2
- Otorgar acceso de red desde un grupo de seguridad de Amazon VPC a la memoria caché
- Descarga y configuración del acceso a la línea de comandos

Cómo crear una Cuenta de AWS

Si no dispone de una Cuenta de AWS, siga estos pasos para crear una.

Procedimiento para registrarse en Cuenta de AWS

1. Abra https://portal.aws.amazon.com/billing/signup.

2. Siga las instrucciones que se le indiquen.

   Parte del procedimiento de registro consiste en recibir una llamada telefónica o mensaje de texto e indicar un código de verificación en el teclado del teléfono.

   Al registrarse en una Cuenta de AWS, se crea un Usuario raíz de la cuenta de AWS. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar tareas que requieren acceso de usuario raíz.

AWS le enviará un correo electrónico de confirmación cuando complete el proceso de registro. Puede ver la actividad de la cuenta y administrarla en cualquier momento entrando en https://aws.amazon.com/ y seleccionando Mi cuenta.

Creación de un usuario con acceso administrativo

Después de registrarse para obtener una Cuenta de AWS, proteja su Usuario raíz de la cuenta de AWS, habilite AWS IAM Identity Center y cree un usuario administrativo para no usar el usuario raíz en las tareas cotidianas.

Protección de Usuario raíz de la cuenta de AWS

1. Inicie sesión en Consola de administración de AWS como propietario de la cuenta; para ello, elija Usuario raíz e introduzca el correo electrónico de su Cuenta de AWS. En la siguiente página, escriba su contraseña.

   Para obtener ayuda para iniciar sesión con el usuario raíz, consulte Iniciar sesión como usuario raíz en la Guía del usuario de AWS Sign-In.
2. Active la autenticación multifactor (MFA) para el usuario raíz.

   Para obtener instrucciones, consulte Habilitación de un dispositivo MFA virtual para su usuario raíz de la Cuenta de AWS (consola) en la Guía del usuario de IAM.

Creación de un usuario con acceso administrativo

1. Activar IAM Identity Center.

   Consulte las instrucciones en Activar AWS IAM Identity Center en la Guía del usuario de AWS IAM Identity Center.

2. En IAM Identity Center, conceda acceso administrativo a un usuario.

   Para ver un tutorial sobre cómo usar Directorio de IAM Identity Center como origen de identidad, consulte Configuración del acceso de los usuarios con el Directorio de IAM Identity Center predeterminado en la Guía del usuario de AWS IAM Identity Center.

Inicio de sesión como usuario con acceso de administrador

- Para iniciar sesión con el usuario de IAM Identity Center, use la URL de inicio de sesión que se envió a la dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

  Para obtener ayuda para iniciar sesión con un usuario de IAM Identity Center, consulte Inicio de sesión en el portal de acceso de AWS en la Guía del usuario de AWS Sign-In.

Concesión de acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos de privilegios mínimos.

   Para conocer las instrucciones, consulte Create a permission set en la Guía del usuario de AWS IAM Identity Center.
2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

   Para conocer las instrucciones, consulte Add groups en la Guía del usuario de AWS IAM Identity Center.

Conceder acceso programático

Los usuarios necesitan acceso programático si desean interactuar con AWS fuera de la Consola de administración de AWS. La forma de conceder el acceso programático depende del tipo de usuario que acceda a AWS.

Para conceder acceso programático a los usuarios, seleccione una de las siguientes opciones.

¿Qué usuario necesita acceso programático?	Para	Mediante
Identidad del personal (Usuarios administrados en el IAM Identity Center)	Utiliza credenciales temporales para firmar las solicitudes programáticas a la AWS CLI, los AWS SDK y las API de AWS.	Siga las instrucciones de la interfaz que desea utilizar: • Para utilizar la AWS CLI, consulte Configuring the AWS CLI to usa AWS IAM Identity Center en la Guía del usuario de AWS Command Line Interface. • Para usar AWS SDK, las herramientas y las API de AWS, consulta IAM Identity Center authentication en la Guía de referencia del SDK y las herramientas de AWS.
IAM	Utiliza credenciales temporales para firmar las solicitudes programáticas a la AWS CLI, los AWS SDK y las API de AWS.	Siguiendo las instrucciones de Uso de credenciales temporales con recursos de AWS de la Guía del usuario de IAM.
IAM	(No recomendado) Utilizar credenciales a largo plazo para firmar las solicitudes programáticas a la AWS CLI, los AWS SDK o las API de AWS.	Siga las instrucciones de la interfaz que desea utilizar: • Para la AWS CLI, consulte Autenticación mediante credenciales de usuario de IAM en la Guía del usuario de AWS Command Line Interface. • Para ver los AWS SDK y las herramientas, consulte Autenticar mediante

¿Qué usuario necesita acceso programático?	Para	Mediante
		credenciales a largo plazo en la Guía de referencia de AWS SDK y herramientas. • Para las API de AWS, consulta Administración de claves de acceso para usuarios de IAM en la Guía del usuario de IAM.

Temas relacionados:

- ¿Qué es IAM? en la Guía del usuario de IAM.
- Credenciales de seguridad de AWS en la Referencia general de AWS.

Configurar los permisos (solo para nuevos usuarios de ElastiCache)

Para dar acceso, agregue permisos a los usuarios, grupos o roles:

- Usuarios y grupos en AWS IAM Identity Center:

  Cree un conjunto de permisos. Siga las instrucciones de Creación de un conjunto de permisos en la Guía del usuario de AWS IAM Identity Center.
- Usuarios gestionados en IAM a través de un proveedor de identidades:

  Cree un rol para la federación de identidades. Siga las instrucciones descritas en Creación de un rol para un proveedor de identidad de terceros (federación) en la Guía del usuario de IAM.
- Usuarios de IAM:
  - Cree un rol que el usuario pueda aceptar. Siga las instrucciones descritas en Creación de un rol para un usuario de IAM en la Guía del usuario de IAM.
  - (No recomendado) Adjunte una política directamente a un usuario o añada un usuario a un grupo de usuarios. Siga las instrucciones descritas en Adición de permisos a un usuario (consola) de la Guía del usuario de IAM.

Amazon ElastiCache crea y utiliza roles vinculados a servicios para aprovisionar recursos y obtener acceso a otros servicios y recursos de AWS en su nombre. Para que ElastiCache cree un rol vinculado a servicios en su nombre, utilice la política administrada por AWS denominada `AmazonElastiCacheFullAccess`. Esta función ya está aprovisionada con los permisos que el servicio requiere para crear un rol vinculado a un servicio en su nombre.

Si lo prefiere, puede no utilizar la política predeterminada, sino una administrada de forma personalizada. En este caso, asegúrese de tener los permisos para llamar a `iam:createServiceLinkedRole`, o bien de haber creado el rol vinculado a servicios de ElastiCache.

Para obtener más información, consulte los siguientes temas:

- Creación de una nueva política (IAM)
- Políticas administradas por AWS para Amazon ElastiCache
- Uso de roles vinculados a servicios para Amazon ElastiCache

Configuración de EC2

Deberá configurar una instancia EC2 desde la que se conectará a la memoria caché.

- Si aún no tiene una instancia de EC2, descubra cómo configurar una instancia de EC2 en Guía de introducción a EC2.
- La instancia de EC2 debe estar en la misma VPC que la memoria caché y debe tener la misma configuración de grupo de seguridad. De forma predeterminada, Amazon ElastiCache crea una memoria caché en la VPC predeterminada y utiliza el grupo de seguridad predeterminado. Para seguir este tutorial, asegúrese de que la instancia EC2 esté en la VPC predeterminada y de que tenga el grupo de seguridad predeterminado.

Otorgar acceso de red desde un grupo de seguridad de Amazon VPC a la memoria caché

Los clústeres basados en nodos de ElastiCache utilizan el puerto 6379 para los comandos de Valkey y Redis OSS. Además, ElastiCache sin servidor utiliza tanto el puerto 6379 como el 6380. Para conectar y ejecutar correctamente los comandos de Valkey o Redis OSS desde su instancia de EC2, su grupo de seguridad debe permitir el acceso a estos puertos de la forma necesaria.

ElastiCache para Memcached utiliza los puertos 11 211 y 11 212 para aceptar los comandos de Memcached. A fin de conectar y ejecutar correctamente los comandos de Memcached desde su instancia EC2, su grupo de seguridad debe permitir el acceso a estos puertos.

1. Inicie sesión en la AWS Command Line Interface y abra la consola de Amazon EC2.
2. En el panel de navegación, en Network & Security (Redes y seguridad), seleccione Security Groups(Grupos de seguridad).
3. En la lista de grupos de seguridad, elija el grupo de seguridad para su Amazon VPC. A menos que haya creado un grupo de seguridad para el uso de ElastiCache, este grupo de seguridad se denominará default (predeterminado).
4. Elija la pestaña Entrada y haga lo siguiente:

    a. Elija Edit (Edición de).

    b. Seleccione Agregar regla.

    c. En la columna Tipo, elija Regla TCP personalizada.

    d. Si utiliza Valkey o Redis OSS, en el cuadro Rango de puertos, escriba `6379`.

    Si usa Memcached, en el cuadro Rango de puertos, escriba `11211`.

    e. En el cuadro Fuente, elija En cualquier lugar que tenga el rango de puertos (0.0.0.0/0), para que cualquier instancia de Amazon EC2 que lance en su Amazon VPC pueda conectarse a la caché.

    f. Si utiliza ElastiCache sin servidor, añada otra regla seleccionando Agregar regla.

    g. En la columna Type, elija Custom TCP rule.

    h. Si utiliza ElastiCache para Redis OSS, en el cuadro Rango de puertos, escriba `6380`.

    Si utiliza ElastiCache para Memcached, en el cuadro Rango de puertos, escriba `11212`.

    i. En el cuadro Fuente, elija En cualquier lugar que tenga el rango de puertos (0.0.0.0/0), para que cualquier instancia de Amazon EC2 que lance en su Amazon VPC pueda conectarse a la caché.

    j. Seleccione Save.

Descarga y configuración del acceso a la línea de comandos

Descargue e instale la utilidad valkey-cli.

Si usa ElastiCache para Valkey, puede que la utilidad valkey-cli le resulte útil. Si utiliza ElastiCache para Redis OSS con redis-cli, considere la posibilidad de cambiar a valkey-cli, ya que también funciona con Redis OSS.

1. Conéctese a la instancia de Amazon EC2 mediante la utilidad de conexión que prefiera. Para obtener instrucciones sobre cómo conectarse a una instancia de Amazon EC2, consulte la Guía de introducción de Amazon EC2.
2. Descargue e instale la utilidad valkey-cli mediante el comando adecuado para su configuración.

   Amazon Linux 2

   ```
   sudo amazon-linux-extras install epel -y
   sudo yum install gcc jemalloc-devel openssl-devel tcl tcl-devel -y
   wget https://github.com/valkey-io/valkey/archive/refs/tags/8.0.0.tar.gz
   tar xvzf valkey-8.0.0.tar.gz
   cd valkey-8.0.0
   make BUILD_TLS=yes
   ```

> **Note**
>
> - Al instalar el paquete redis6, se instalará redis6-cli con soporte de cifrado predeterminado.
> - Es importante contar con soporte de compilación para TLS al instalar valkey-cli o redis-cli. Solo se puede acceder a ElastiCache Serverless cuando TLS está habilitado.
> - Si el clúster al que se conecta no está cifrado, no necesita la opción `Build_TLS=yes`.

Creación de una caché sin servidor de Valkey

En este paso, creará una nueva caché en Amazon ElastiCache.

Consola de administración de AWS

Para crear una nueva caché con la consola de ElastiCache:

1. Inicie sesión en la Consola de administración de AWS y abra https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación del lado izquierdo de la consola, elija Cachés de Valkey.

3. En el lado derecho de la consola, elija Crear caché de Valkey.
4. En Configuración de caché, introduzca un Nombre. Si lo desea, puede especificar una descripción para la caché.
5. Deje seleccionado el ajuste predeterminado.
6. Haga clic en Crear para crear la caché.
7. Cuando la caché se encuentre en estado “ACTIVO”, podrá empezar a usarla para leer y escribir.

AWS CLI

En el siguiente ejemplo de la AWS CLI, se crea una nueva caché mediante create-serverless-cache.

Linux

```
aws elasticache create-serverless-cache \
    --serverless-cache-name CacheName \
    --engine valkey
```

Windows

```
aws elasticache create-serverless-cache ^
    --serverless-cache-name CacheName ^
    --engine valkey
```

Como puede ver, el valor del campo Estado es `CREATING`.

Para comprobar que ElastiCache haya terminado de crear la caché, utilice el comando `describe-serverless-caches`.

Linux

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Después de crear la caché, continúe con Lectura y escritura de datos en la caché.

Lectura y escritura de datos en la caché

En esta sección se da por sentado que ha creado una instancia de Amazon EC2 y que se puede conectar a ella. Para obtener instrucciones al respecto, consulte la Guía de introducción a Amazon EC2.

En esta sección, también se supone que ha configurado los ajustes de acceso a la VPC y del grupo de seguridad para la instancia de EC2 desde la que se conecta a la caché y que ha configurado valkey-cli en la instancia de EC2. Para obtener más información sobre este paso, consulte Configuración de ElastiCache.

Además de los pasos que se indican a continuación, si tiene una aplicación grande o global, puede aumentar considerablemente el rendimiento de lectura creando réplicas y leyéndolas. Para obtener más información sobre este paso más avanzado, consulte Prácticas recomendadas para utilizar réplicas de lectura.

Buscar el punto de conexión de la caché

Consola de administración de AWS

Para buscar el punto de conexión de la caché con la consola de ElastiCache:

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación del lado izquierdo de la consola, elija Cachés de Valkey.
3. En la parte derecha de la consola, haga clic en el nombre de la caché que acaba de crear.
4. En los Detalles de la caché, busque y copie el punto de conexión de la caché.

AWS CLI

En el siguiente ejemplo de la AWS CLI, se muestra cómo encontrar el punto de conexión de la nueva caché mediante el comando describe-serverless-caches. Cuando haya ejecutado el comando, busque el campo “Punto de conexión”.

Linux

```
aws elasticache describe-serverless-caches \
  --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches ^
  --serverless-cache-name CacheName
```

Conexión a la caché de Valkey (Linux)

Ahora que tiene el punto de conexión que necesita, puede iniciar sesión en una instancia EC2 y conectarse a la caché. En el siguiente ejemplo, se conecta a un clúster con la utilidad valkey-cli. El siguiente comando se conecta a una caché (nota: sustituya cache-endpoint por el punto de conexión que recuperó en el paso anterior).

```
src/valkey-cli -h cache-endpoint --tls -p 6379
set a "hello"          // Set key "a" with a string value and no expiration
OK
get a                  // Get value for key "a"
"hello"
```

Conexión a la caché de Valkey (Windows)

Ahora que tiene el punto de conexión que necesita, puede iniciar sesión en una instancia EC2 y conectarse a la caché. En el siguiente ejemplo, se conecta a un clúster con la utilidad valkey-cli. El siguiente comando se conecta a una caché. Abra el símbolo del sistema, vaya al directorio de Valkey o Redis OSS y ejecute el comando (nota: sustituya Cache_Endpoint por el punto de conexión que recuperó en el paso anterior).

```
c:\Valkey>valkey-cli -h Valkey_Cluster_Endpoint --tls -p 6379
set a "hello"          // Set key "a" with a string value and no expiration
OK
get a                  // Get value for key "a"
"hello"
```

Ahora puede pasar a (Opcional) Limpieza.

(Opcional) Limpieza

Si ya no necesita la caché de Amazon ElastiCache que creó, puede eliminarla. De este modo, no se le cobrará por recursos que no vaya a utilizar. Para eliminar la caché, puede utilizar la consola de ElastiCache, la AWS CLI o la API de ElastiCache.

Consola de administración de AWS

Para eliminar la caché con la consola:

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación del lado izquierdo de la consola, elija Cachés de Valkey.
3. Elija el botón de opción situado al lado de la caché que desea eliminar.
4. Seleccione Acciones, en la parte superior derecha, y Eliminar.
5. Si lo desea, puede tomar una última instantánea antes de eliminar la memoria caché.
6. En la pantalla de confirmación Eliminar, vuelva a introducir el nombre de la caché y seleccione Eliminar, para eliminar el clúster, o Cancelar, para conservarlo.

En cuanto la caché pase al estado DELETING, dejará de generar gastos.

AWS CLI

En el siguiente ejemplo de la AWS CLI, se elimina una caché mediante el comando delete-serverless-cache.

Linux

```
aws elasticache delete-serverless-cache \
  --serverless-cache-name CacheName
```

Windows

```
aws elasticache delete-serverless-cache ^
  --serverless-cache-name CacheName
```

Como puede ver, el valor del campo Estado es DELETING.

Ahora puede pasar a Siguientes pasos.

Siguientes pasos

Para obtener más información sobre ElastiCache, consulte las siguientes páginas:

- Uso de ElastiCache
- Escalado de ElastiCache

- Registro y Supervisión en Amazon ElastiCache
- Procedimientos recomendados y estrategias de almacenamiento en caché de ElastiCache
- Instantánea y restauración
- Supervisión de eventos de ElastiCache en Amazon SNS

Creación de una caché sin servidor de Redis OSS

En este paso, creará una nueva caché en Amazon ElastiCache.

Consola de administración de AWS

Para crear una nueva caché con la consola de ElastiCache:

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación del lado izquierdo de la consola, elija Cachés de Redis OSS.
3. En el lado derecho de la consola, elija Crear una caché de Redis OSS.
4. En Configuración de caché, introduzca un Nombre. Si lo desea, puede especificar una descripción para la caché.
5. Deje seleccionado el ajuste predeterminado.
6. Haga clic en Crear para crear la caché.
7. Cuando la caché se encuentre en estado “ACTIVO”, podrá empezar a usarla para leer y escribir.

AWS CLI

En el siguiente ejemplo de la AWS CLI, se crea una nueva caché mediante create-serverless-cache.

Linux

```
aws elasticache create-serverless-cache \
    --serverless-cache-name CacheName \
    --engine redis
```

Windows

```
aws elasticache create-serverless-cache ^
    --serverless-cache-name CacheName ^
```

```
    --engine redis
```

Como puede ver, el valor del campo Estado es `CREATING`.

Para comprobar que ElastiCache haya terminado de crear la caché, utilice el comando `describe-serverless-caches`.

Linux

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Después de crear la caché, continúe con Lectura y escritura de datos en la caché.

Lectura y escritura de datos en la caché

En esta sección se da por sentado que ha creado una instancia de Amazon EC2 y que se puede conectar a ella. Para obtener instrucciones al respecto, consulte la Guía de introducción a Amazon EC2.

En esta sección, también se supone que ha configurado los ajustes de acceso a la VPC y del grupo de seguridad para la instancia de EC2 desde la que se conecta a la caché y que ha configurado valkey-cli en la instancia de EC2. Para obtener más información sobre este paso, consulte Configuración de ElastiCache.

Buscar el punto de conexión de la caché

Consola de administración de AWS

Para buscar el punto de conexión de la caché con la consola de ElastiCache:

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación del lado izquierdo de la consola, elija Cachés de Valkey o Cachés de Redis OSS.
3. En la parte derecha de la consola, haga clic en el nombre de la caché que acaba de crear.
4. En los Detalles de la caché, busque y copie el punto de conexión de la caché.

AWS CLI

En el siguiente ejemplo de la AWS CLI, se muestra cómo encontrar el punto de conexión de la nueva caché mediante el comando describe-serverless-caches. Cuando haya ejecutado el comando, busque el campo “Punto de conexión”.

Linux

```
aws elasticache describe-serverless-caches \
  --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches ^
  --serverless-cache-name CacheName
```

Conexión a la caché de Valkey o Redis OSS (Linux)

Ahora que tiene el punto de conexión que necesita, puede iniciar sesión en una instancia EC2 y conectarse a la caché. En el siguiente ejemplo, se conecta a un clúster con la utilidad valkey-cli. El siguiente comando se conecta a una caché (nota: sustituya cache-endpoint por el punto de conexión que recuperó en el paso anterior).

```
src/valkey-cli -h cache-endpoint --tls -p 6379
set a "hello"          // Set key "a" with a string value and no expiration
OK
get a                  // Get value for key "a"
"hello"
```

Conexión a la caché de Valkey o Redis OSS (Windows)

Ahora que tiene el punto de conexión que necesita, puede iniciar sesión en una instancia EC2 y conectarse a la caché. En el siguiente ejemplo, se conecta a un clúster con la utilidad valkey-cli. El siguiente comando se conecta a una caché. Abra el símbolo del sistema, vaya al directorio de Valkey y ejecute el comando (nota: sustituya Cache_Endpoint por el punto de conexión que recuperó en el paso anterior).

```
c:\Redis>valkey-cli -h Redis_Cluster_Endpoint --tls -p 6379
set a "hello"          // Set key "a" with a string value and no expiration
OK
get a                  // Get value for key "a"
```

```
"hello"
```

Ahora puede pasar a (Opcional) Limpieza.

(Opcional) Limpieza

Si ya no necesita la caché de Amazon ElastiCache que creó, puede eliminarla. De este modo, no se le cobrará por recursos que no vaya a utilizar. Para eliminar la caché, puede utilizar la consola de ElastiCache, la AWS CLI o la API de ElastiCache.

Consola de administración de AWS

Para eliminar la caché con la consola:

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación de la parte izquierda de la consola, elija Cachés de Valkey o Redis OSS.
3. Elija el botón de opción situado al lado de la caché que desea eliminar.
4. Seleccione Acciones, en la parte superior derecha, y Eliminar.
5. Si lo desea, puede tomar una última instantánea antes de eliminar la memoria caché.
6. En la pantalla de confirmación Eliminar, vuelva a introducir el nombre de la caché y seleccione Eliminar, para eliminar el clúster, o Cancelar, para conservarlo.

En cuanto la caché pase al estado DELETING, dejará de generar gastos.

AWS CLI

En el siguiente ejemplo de la AWS CLI, se elimina una caché mediante el comando delete-serverless-cache.

Linux

```
aws elasticache delete-serverless-cache \
  --serverless-cache-name CacheName
```

Windows

```
aws elasticache delete-serverless-cache ^
```

```
  --serverless-cache-name CacheName
```

Como puede ver, el valor del campo Estado es DELETING.

Ahora puede pasar a Siguientes pasos.

Siguientes pasos

Para obtener más información sobre ElastiCache, consulte las siguientes páginas:

- Uso de ElastiCache
- Escalado de ElastiCache
- Registro y Supervisión en Amazon ElastiCache
- Procedimientos recomendados y estrategias de almacenamiento en caché de ElastiCache
- Instantánea y restauración
- Supervisión de eventos de ElastiCache en Amazon SNS

Creación de una caché sin servidor de Memcached

Consola de administración de AWS

Para crear una nueva caché sin servidor de Memcached con la consola de ElastiCache:

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación del lado izquierdo de la consola, elija Cachés de Memcached.
3. En el lado derecho de la consola, elija Crear caché de Memcached.
4. En Configuración de caché, introduzca un Nombre. Si lo desea, puede especificar una descripción para la caché.
5. Deje seleccionado el ajuste predeterminado.
6. Haga clic en Crear para crear la caché.
7. Una vez que la caché esté en estado “ACTIVO”, puede empezar a escribir y leer datos en ella.

Para crear una nueva caché con la AWS CLI

En el siguiente ejemplo de la AWS CLI, se crea una nueva caché mediante create-serverless-cache.

Linux

```
aws elasticache create-serverless-cache \
  --serverless-cache-name CacheName \
  --engine memcached
```

Windows

```
aws elasticache create-serverless-cache ^
  --serverless-cache-name CacheName ^
  --engine memcached
```

Como puede ver, el valor del campo Estado es `CREATING`.

Para comprobar que ElastiCache haya terminado de crear la caché, utilice el comando `describe-serverless-caches`.

Linux

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Después de crear la caché, continúe con Lectura y escritura de datos en la caché.

Lectura y escritura de datos en la caché

En esta sección se da por sentado que ha creado una instancia de Amazon EC2 y que se puede conectar a ella. Para obtener instrucciones al respecto, consulte la Guía de introducción a Amazon EC2.

De forma predeterminada, ElastiCache crea una caché de en su VPC predeterminada. Asegúrese de que la instancia EC2 también se cree en la VPC predeterminada para que pueda conectarse a la caché.

Buscar el punto de conexión de la caché

Consola de administración de AWS

Para buscar el punto de conexión de la caché con la consola de ElastiCache:

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación del lado izquierdo de la consola, elija Cachés de Memcached.
3. En la parte derecha de la consola, haga clic en el nombre de la caché que acaba de crear.
4. En los Detalles de la caché, busque y copie el punto de conexión de la caché.

AWS CLI

En el siguiente ejemplo de la AWS CLI, se muestra cómo encontrar el punto de conexión de la nueva caché mediante el comando describe-serverless-caches. Cuando haya ejecutado el comando, busque el campo “Punto de conexión”.

Linux

```
aws elasticache describe-serverless-caches \
  --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches ^
  --serverless-cache-name CacheName
```

Conexión mediante OpenSSL

Para obtener información sobre cómo conectarse empleando OpenSSL, consulte Cifrado en tránsito (TLS) de ElastiCache.

Conexión mediante el cliente Java de Memcached

```
import java.security.KeyStore;
import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManagerFactory;
import net.spy.memcached.AddrUtil;
import net.spy.memcached.ConnectionFactoryBuilder;
import net.spy.memcached.FailureMode;
import net.spy.memcached.MemcachedClient;

public class TLSDemo {
```

```
    public static void main(String[] args) throws Exception {
        ConnectionFactoryBuilder connectionFactoryBuilder = new
 ConnectionFactoryBuilder();
        // Build SSLContext
        TrustManagerFactory tmf =
 TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm());
        tmf.init((KeyStore) null);
        SSLContext sslContext = SSLContext.getInstance("TLS");
        sslContext.init(null, tmf.getTrustManagers(), null);
        // Create the client in TLS mode
        connectionFactoryBuilder.setSSLContext(sslContext);
        // Set Failure Mode to Retry
        connectionFactoryBuilder.setFailureMode(FailureMode.Retry);
        MemcachedClient client = new MemcachedClient(connectionFactoryBuilder.build(),
 AddrUtil.getAddresses("mycluster-fnjyzo.serverless.use1.cache.amazonaws.com:11211"));

        // Store a data item for an hour.
        client.set("theKey", 3600, "This is the data value");
    }
}
```

Conexión mediante el cliente PHP de Memcached

```
<?php
$cluster_endpoint = "mycluster.serverless.use1.cache.amazonaws.com";
$server_port = 11211;

/* Initialize a persistent Memcached client in TLS mode */
$tls_client = new Memcached('persistent-id');
$tls_client->addServer($cluster_endpoint, $server_port);
if(!$tls_client->setOption(Memcached::OPT_USE_TLS, 1)) {
    echo $tls_client->getLastErrorMessage(), "\n";
    exit(1);
}
$tls_config = new MemcachedTLSContextConfig();
$tls_config->hostname = '*.serverless.use1.cache.amazonaws.com';
$tls_config->skip_cert_verify = false;
$tls_config->skip_hostname_verify = false;
$tls_client->createAndSetTLSContext((array)$tls_config);

 /* store the data for 60 seconds in the cluster */
$tls_client->set('key', 'value', 60);
?>
```

Conexión mediante el cliente Python de Memcached (Pymemcache)

Consulte https://pymemcache.readthedocs.io/en/latest/getting_started.html

```
import ssl
from pymemcache.client.base import Client

context = ssl.create_default_context()
cluster_endpoint = <To be taken from the AWS CLI / console>
target_port = 11211
memcached_client = Client(("{cluster_endpoint}", target_port), tls_context=context)
memcached_client.set("key", "value", expire=500, noreply=False)
assert self.memcached_client.get("key").decode() == "value"
```

Conexión mediante el cliente NodeJS/TS de Memcached (Electrode-IO memcache)

Consulte https://github.com/electrode-io/memcache y https://www.npmjs.com/package/memcache-client

Instalación mediante `npm i memcache-client`

En la aplicación, cree un cliente TLS de memcached de la siguiente manera:

```
var memcache = require("memcache-client");
const client = new memcache.MemcacheClient({server: "{cluster_endpoint}:11211", tls:
 {}});
client.set("key", "value");
```

Conexión mediante el cliente Rust de Memcached (rust-memcache)

Consulte https://crates.io/crates/memcache y https://github.com/aisk/rust-memcache.

```
// create connection with to memcached server node:
let client = memcache::connect("memcache+tls://<cluster_endpoint>:11211?
verify_mode=none").unwrap();

// set a string value
client.set("foo", "bar", 0).unwrap();
```

Conexión mediante el cliente Go de Memcached (Gomemcache)

Consulte https://github.com/bradfitz/gomemcache

```
c := New(net.JoinHostPort("{cluster_endpoint}", strconv.Itoa(port)))
c.DialContext = func(ctx context.Context, network, addr string) (net.Conn, error) {
var td tls.Dialer
td.Config = &tls.Config{}
return td.DialContext(ctx, network, addr)
}
foo := &Item{Key: "foo", Value: []byte("fooval"), Flags: 123}
err := c.Set(foo)
```

Conexión mediante el cliente Ruby de Memcached (Dalli)

Consulte https://github.com/petergoldstein/dalli

```
require 'dalli'
ssl_context = OpenSSL::SSL::SSLContext.new
ssl_context.ssl_version = :SSLv23
ssl_context.verify_hostname = true
ssl_context.verify_mode = OpenSSL::SSL::VERIFY_PEER
client = Dalli::Client.new("<cluster_endpoint>:11211", :ssl_context => ssl_context);
client.get("abc")
```

Conexión mediante el cliente .NET de Memcached (EnyimMemcachedCore)

Consulte https://github.com/cnblogs/EnyimMemcachedCore

```
"MemcachedClient": {
"Servers": [
{
"Address": "{cluster_endpoint}",
"Port": 11211
}
],
"UseSslStream":  true
}
```

Ahora puede pasar a (Opcional) Limpieza.

(Opcional) Limpieza

Uso de Consola de administración de AWS

El siguiente procedimiento elimina una sola caché de su implementación. Para eliminar varias cachés, repita el procedimiento por cada caché que desee eliminar. No es necesario esperar a que una caché se elimine por completo antes de empezar a eliminar otra.

Para eliminar una caché

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de la consola de ElastiCache, elija el motor en el que se ejecuta la caché que desea eliminar. Se mostrará una lista de todas las cachés que ejecutan ese motor.
3. Para elegir la caché que se va a eliminar, elija el nombre de la caché en la lista.

   > **Important**
   >
   > Las cachés solo se pueden eliminar una por una en la consola de ElastiCache. Al elegir varias cachés, se deshabilita la operación de eliminación.

4. En Actions (Acciones), seleccione Delete (Eliminar).
5. En la pantalla de confirmación Eliminar caché, elija Eliminar para eliminar la caché o elija Cancelar para conservarla.
6. Si elige Eliminar, el estado de la caché cambiará a eliminando.

En cuanto la caché pase al estado DELETING, dejará de generar gastos.

Uso de AWS CLI

El siguiente código elimina my-cache de la caché.

```
aws elasticache delete-serverless-cache --serverless-cache-name my-cache
```

La acción delete-serverless-cache de la CLI solo elimina una caché sin servidor. Para eliminar varias cachés, llame a delete-serverless-cache para cada caché sin servidor que desee eliminar. No es necesario esperar a que una caché sin servidor acabe de eliminarse antes de eliminar otra.

Para Linux, macOS o Unix:

```
aws elasticache delete-serverless-cache \
  --serverless-cache-name my-cache
```

Para Windows:

```
aws elasticache delete-serverless-cache ^
  --serverless-cache-name my-cache
```

Para obtener más información, consulte el tema delete-serverless-cache de AWS CLI para ElastiCache.

Ahora puede pasar a Siguientes pasos.

Siguientes pasos

Para obtener más información sobre ElastiCache, consulte

- Uso de ElastiCache
- Escalado de ElastiCache
- Cuotas de ElastiCache
- Procedimientos recomendados y estrategias de almacenamiento en caché de ElastiCache
- Visualización de eventos de ElastiCache

Tutoriales: introducción a Python y ElastiCache

Esta sección contiene tutoriales prácticos para ayudar a entender ElastiCache para Valkey y Redis OSS. Recomendamos que realice alguno de los tutoriales específicos de los distintos lenguajes.

> **Note**
>
> AWSLos SDK de están disponibles en una amplia variedad de idiomas. Para obtener una lista completa, consulte Herramientas para Amazon Web Services.

Temas

- Python y ElastiCache

Python y ElastiCache

En este tutorial, utilizaremos AWS SDK for Python (Boto3) para escribir programas sencillos que lleven a cabo las siguientes operaciones de ElastiCache:

- Crear clústeres de ElastiCache para Redis OSS (modo de clúster habilitado y modo de clúster deshabilitado).
- Verificar si los usuarios o grupos de usuarios existen; de lo contrario, crearlos. (Esta característica está disponible con Valkey 7.2 en adelante y con Redis OSS 6.0 a 7.1).
- Conectarse a ElastiCache
- Realice operaciones tales como configurar y obtener cadenas, leer y escribir en flujos, y publicar y suscribirse desde el canal Pub/Sub.

Mientras sigue este tutorial, puede consultar la documentación de AWS SDK for Python (Boto) La siguiente sección es específica de ElastiCache: Cliente de nivel bajo de ElastiCache

Requisitos previos del tutorial

- Configure una clave de acceso de AWS para utilizar los SDK de AWS. Para obtener más información, consulte Configuración de ElastiCache.
- Instale Python 3.0 o una versión posterior. Para obtener más información, consulte https://www.python.org/downloads. Para obtener instrucciones, consulte Quickstart en la documentación de Boto 3.

Temas

- Tutorial: creación de usuarios y clústeres de ElastiCache
- Tutorial: conexión a ElastiCache
- Ejemplos de uso

Tutorial: creación de usuarios y clústeres de ElastiCache

En los siguientes ejemplos, se utiliza el SDK de boto3 para las operaciones de administración de ElastiCache para Redis OSS (creación de clústeres o usuarios) y redis-py/redis-py-cluster para la gestión de los datos.

Temas

- Crear un clúster en modo de clúster deshabilitado
- Crear un clúster en modo de clúster deshabilitado con TLS y RBAC
- Crear un clúster en modo de clúster habilitado
- Crear un clúster en modo de clúster habilitado con TLS y RBAC
- Verificar si los usuarios o grupos de usuarios existen, de lo contrario, es necesario crearlos

Crear un clúster en modo de clúster deshabilitado

Copie el siguiente programa y péguelo en un archivo denominado CreateClusterModeDisabledCluster.py.

```
import boto3
import logging

logging.basicConfig(level=logging.INFO)
client = boto3.client('elasticache')

def
 create_cluster_mode_disabled(CacheNodeType='cache.t3.small',EngineVersion='6.0',NumCacheCluste
 cluster',ReplicationGroupId=None):
    """Creates an ElastiCache Cluster with cluster mode disabled

    Returns a dictionary with the API response

    :param CacheNodeType: Node type used on the cluster. If not specified,
 cache.t3.small will be used
    Refer to https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/
CacheNodes.SupportedTypes.html for supported node types
    :param EngineVersion: Engine version to be used. If not specified, latest will be
 used.
    :param NumCacheClusters: Number of nodes in the cluster. Minimum 1 (just a primary
 node) and maximun 6 (1 primary and 5 replicas).
    If not specified, cluster will be created with 1 primary and 1 replica.
    :param ReplicationGroupDescription: Description for the cluster.
    :param ReplicationGroupId: Name for the cluster
    :return: dictionary with the API results

    """
    if not ReplicationGroupId:
        return 'ReplicationGroupId parameter is required'
```

```
    response = client.create_replication_group(
        AutomaticFailoverEnabled=True,
        CacheNodeType=CacheNodeType,
        Engine='valkey',
        EngineVersion=EngineVersion,
        NumCacheClusters=NumCacheClusters,
        ReplicationGroupDescription=ReplicationGroupDescription,
        ReplicationGroupId=ReplicationGroupId,
        SnapshotRetentionLimit=30,
    )
    return response


if __name__ == '__main__':

    # Creates an ElastiCache Cluster mode disabled cluster, based on cache.m6g.large
 nodes, Valkey 8.0, one primary and two replicas
    elasticacheResponse = create_cluster_mode_disabled(
        #CacheNodeType='cache.m6g.large',
        EngineVersion='8.0',
        NumCacheClusters=3,
        ReplicationGroupDescription='Valkey cluster mode disabled with replicas',
        ReplicationGroupId='valkey202104053'
        )

    logging.info(elasticacheResponse)
```

Para ejecutar el programa, introduzca el siguiente comando:

```
python CreateClusterModeDisabledCluster.py
```

Para obtener más información, consulte Administración de clústeres en ElastiCache.

Crear un clúster en modo de clúster deshabilitado con TLS y RBAC

Para garantizar la seguridad, puede utilizar Transport Layer Security (TLS) y el control de acceso basado en roles (RBAC) al crear un clúster en modo de clúster deshabilitado. A diferencia de AUTH de Valkey o Redis OSS, donde todos los clientes autenticados tienen acceso completo al grupo de replicación si su token se encuentra autenticado, RBAC permite controlar el acceso al clúster a través de grupos de usuarios. Estos grupos de usuarios se han diseñado como una forma de organizar el acceso a los grupos de reproducción. Para obtener más información, consulte Control de acceso basado en roles (RBAC).

Copie el siguiente programa y péguelo en un archivo denominado ClusterModeDisabledWithRBAC.py.

```
import boto3
import logging

logging.basicConfig(level=logging.INFO)
client = boto3.client('elasticache')

def
 create_cluster_mode_disabled_rbac(CacheNodeType='cache.t3.small',EngineVersion='6.0',NumCacheC
 cluster',ReplicationGroupId=None, UserGroupIds=None,
 SecurityGroupIds=None,CacheSubnetGroupName=None):
    """Creates an ElastiCache Cluster with cluster mode disabled and RBAC

    Returns a dictionary with the API response

    :param CacheNodeType: Node type used on the cluster. If not specified,
 cache.t3.small will be used
    Refer to https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/
CacheNodes.SupportedTypes.html for supported node types
    :param EngineVersion: Engine version to be used. If not specified, latest will be
 used.
    :param NumCacheClusters: Number of nodes in the cluster. Minimum 1 (just a primary
 node) and maximun 6 (1 primary and 5 replicas).
    If not specified, cluster will be created with 1 primary and 1 replica.
    :param ReplicationGroupDescription: Description for the cluster.
    :param ReplicationGroupId: Mandatory name for the cluster.
    :param UserGroupIds: The ID of the user group to be assigned to the cluster.
    :param SecurityGroupIds: List of security groups to be assigned. If not defined,
 default will be used
    :param CacheSubnetGroupName: subnet group where the cluster will be placed. If not
 defined, default will be used.
    :return: dictionary with the API results

    """
    if not ReplicationGroupId:
        return {'Error': 'ReplicationGroupId parameter is required'}
    elif not isinstance(UserGroupIds,(list)):
        return {'Error': 'UserGroupIds parameter is required and must be a list'}

    params={'AutomaticFailoverEnabled': True,
            'CacheNodeType': CacheNodeType,
```

```
            'Engine': 'valkey',
            'EngineVersion': EngineVersion,
            'NumCacheClusters': NumCacheClusters,
            'ReplicationGroupDescription': ReplicationGroupDescription,
            'ReplicationGroupId': ReplicationGroupId,
            'SnapshotRetentionLimit': 30,
            'TransitEncryptionEnabled': True,
            'UserGroupIds':UserGroupIds
        }

    # defaults will be used if CacheSubnetGroupName or SecurityGroups are not explicit.
    if isinstance(SecurityGroupIds,(list)):
        params.update({'SecurityGroupIds':SecurityGroupIds})
    if CacheSubnetGroupName:
        params.update({'CacheSubnetGroupName':CacheSubnetGroupName})

    response = client.create_replication_group(**params)
    return response

if __name__ == '__main__':

    # Creates an ElastiCache Cluster mode disabled cluster, based on cache.m6g.large
 nodes, Valkey 8.0, one primary and two replicas.
    # Assigns the existent user group "mygroup" for RBAC authentication

    response=create_cluster_mode_disabled_rbac(
        CacheNodeType='cache.m6g.large',
        EngineVersion='8.0',
        NumCacheClusters=3,
        ReplicationGroupDescription='Valkey cluster mode disabled with replicas',
        ReplicationGroupId='valkey202104',
        UserGroupIds=[
            'mygroup'
        ],
        SecurityGroupIds=[
            'sg-7cc73803'
        ],
        CacheSubnetGroupName='default'
    )

    logging.info(response)
```

Para ejecutar el programa, introduzca el siguiente comando:

```
python ClusterModeDisabledWithRBAC.py
```

Para obtener más información, consulte Administración de clústeres en ElastiCache.

Crear un clúster en modo de clúster habilitado

Copie el siguiente programa y péguelo en un archivo denominado ClusterModeEnabled.py.

```
import boto3
import logging

logging.basicConfig(level=logging.INFO)
client = boto3.client('elasticache')

def
 create_cluster_mode_enabled(CacheNodeType='cache.t3.small',EngineVersion='6.0',NumNodeGroups=1
 ReplicationGroupDescription='Sample cache with cluster mode
 enabled',ReplicationGroupId=None):
    """Creates an ElastiCache Cluster with cluster mode enabled

    Returns a dictionary with the API response

    :param CacheNodeType: Node type used on the cluster. If not specified,
 cache.t3.small will be used
    Refer to https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/
CacheNodes.SupportedTypes.html for supported node types
    :param EngineVersion: Engine version to be used. If not specified, latest will be
 used.
    :param NumNodeGroups: Number of shards in the cluster. Minimum 1 and maximun 90.
    If not specified, cluster will be created with 1 shard.
    :param ReplicasPerNodeGroup: Number of replicas per shard. If not specified 1
 replica per shard will be created.
    :param ReplicationGroupDescription: Description for the cluster.
    :param ReplicationGroupId: Name for the cluster
    :return: dictionary with the API results

    """
    if not ReplicationGroupId:
        return 'ReplicationGroupId parameter is required'

    response = client.create_replication_group(
        AutomaticFailoverEnabled=True,
        CacheNodeType=CacheNodeType,
        Engine='valkey',
```

```
        EngineVersion=EngineVersion,
        ReplicationGroupDescription=ReplicationGroupDescription,
        ReplicationGroupId=ReplicationGroupId,
    #   Creates a cluster mode enabled cluster with 1 shard(NumNodeGroups), 1 primary
 node (implicit) and 2 replicas (replicasPerNodeGroup)
        NumNodeGroups=NumNodeGroups,
        ReplicasPerNodeGroup=ReplicasPerNodeGroup,
        CacheParameterGroupName='default.valkey7.2.cluster.on'
    )

    return response


# Creates a cluster mode enabled
response = create_cluster_mode_enabled(
    CacheNodeType='cache.m6g.large',
    EngineVersion='6.0',
    ReplicationGroupDescription='Valkey cluster mode enabled with replicas',
    ReplicationGroupId='valkey20210',
#   Creates a cluster mode enabled cluster with 1 shard(NumNodeGroups), 1 primary
 (implicit) and 2 replicas (replicasPerNodeGroup)
    NumNodeGroups=2,
    ReplicasPerNodeGroup=1,
)

logging.info(response)
```

Para ejecutar el programa, introduzca el siguiente comando:

```
python ClusterModeEnabled.py
```

Para obtener más información, consulte Administración de clústeres en ElastiCache.

Crear un clúster en modo de clúster habilitado con TLS y RBAC

Para garantizar la seguridad, puede utilizar Transport Layer Security (TLS) y el control de acceso basado en roles (RBAC) al crear un clúster en modo de clúster habilitado. A diferencia de AUTH de Valkey o Redis OSS, donde todos los clientes autenticados tienen acceso completo al grupo de replicación si su token se encuentra autenticado, RBAC permite controlar el acceso al clúster a través de grupos de usuarios. Estos grupos de usuarios se han diseñado como una forma de organizar el acceso a los grupos de reproducción. Para obtener más información, consulte Control de acceso basado en roles (RBAC).

Copie el siguiente programa y péguelo en un archivo denominado ClusterModeEnabledWithRBAC.py.

```
import boto3
import logging

logging.basicConfig(level=logging.INFO)
client = boto3.client('elasticache')

def
 create_cluster_mode_enabled(CacheNodeType='cache.t3.small',EngineVersion='6.0',NumNodeGroups=1
 ReplicationGroupDescription='Sample cache with cluster
 mode enabled',ReplicationGroupId=None,UserGroupIds=None,
 SecurityGroupIds=None,CacheSubnetGroupName=None,CacheParameterGroupName='default.valkey7.2.clu
    """Creates an ElastiCache Cluster with cluster mode enabled and RBAC

    Returns a dictionary with the API response

    :param CacheNodeType: Node type used on the cluster. If not specified,
 cache.t3.small will be used
    Refer to https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/
CacheNodes.SupportedTypes.html for supported node types
    :param EngineVersion: Engine version to be used. If not specified, latest will be
 used.
    :param NumNodeGroups: Number of shards in the cluster. Minimum 1 and maximun 90.
    If not specified, cluster will be created with 1 shard.
    :param ReplicasPerNodeGroup: Number of replicas per shard. If not specified 1
 replica per shard will be created.
    :param ReplicationGroupDescription: Description for the cluster.
    :param ReplicationGroupId: Name for the cluster.
    :param CacheParameterGroupName: Parameter group to be used. Must be compatible with
 the engine version and cluster mode enabled.
    :return: dictionary with the API results

    """
    if not ReplicationGroupId:
        return 'ReplicationGroupId parameter is required'
    elif not isinstance(UserGroupIds,(list)):
        return {'Error': 'UserGroupIds parameter is required and must be a list'}

    params={'AutomaticFailoverEnabled': True,
            'CacheNodeType': CacheNodeType,
            'Engine': 'valkey',
```

```
            'EngineVersion': EngineVersion,
            'ReplicationGroupDescription': ReplicationGroupDescription,
            'ReplicationGroupId': ReplicationGroupId,
            'SnapshotRetentionLimit': 30,
            'TransitEncryptionEnabled': True,
            'UserGroupIds':UserGroupIds,
            'NumNodeGroups': NumNodeGroups,
            'ReplicasPerNodeGroup': ReplicasPerNodeGroup,
            'CacheParameterGroupName': CacheParameterGroupName
        }

    # defaults will be used if CacheSubnetGroupName or SecurityGroups are not explicit.
    if isinstance(SecurityGroupIds,(list)):
        params.update({'SecurityGroupIds':SecurityGroupIds})
    if CacheSubnetGroupName:
        params.update({'CacheSubnetGroupName':CacheSubnetGroupName})

    response = client.create_replication_group(**params)
    return response

if __name__ == '__main__':
    # Creates a cluster mode enabled cluster
    response = create_cluster_mode_enabled(
        CacheNodeType='cache.m6g.large',
        EngineVersion='7.2',
        ReplicationGroupDescription='Valkey cluster mode enabled with replicas',
        ReplicationGroupId='valkey2021',
    #   Creates a cluster mode enabled cluster with 1 shard(NumNodeGroups), 1 primary
 (implicit) and 2 replicas (replicasPerNodeGroup)
        NumNodeGroups=2,
        ReplicasPerNodeGroup=1,
        UserGroupIds=[
            'mygroup'
        ],
        SecurityGroupIds=[
            'sg-7cc73803'
        ],
        CacheSubnetGroupName='default'

    )

    logging.info(response)
```

Para ejecutar el programa, introduzca el siguiente comando:

```
python ClusterModeEnabledWithRBAC.py
```

Para obtener más información, consulte Administración de clústeres en ElastiCache.

Verificar si los usuarios o grupos de usuarios existen, de lo contrario, es necesario crearlos

Con el RBAC, puede crear usuarios y asignarles permisos específicos mediante una cadena de acceso. Asigne los usuarios a grupos de usuarios alineados con un rol específico (administradores y personal de Recursos Humanos) que luego se implementan en uno o más grupos de replicación de ElastiCache para Redis OSS. Al hacerlo, puede establecer límites de seguridad entre clientes que utilicen el mismo grupo o grupos de replicación de Valkey o Redis OSS e impedir que los clientes accedan a los datos de los demás. Para obtener más información, consulte Control de acceso basado en roles (RBAC).

Copie el siguiente programa y péguelo en un archivo denominado UserAndUserGroups.py. Actualice el mecanismo de suministro de credenciales. Las credenciales de este ejemplo se muestran como reemplazables y se les asigna un elemento no declarado. Evite codificar las credenciales de forma rígida.

En este ejemplo, se utiliza una cadena de acceso con los permisos del usuario. Para obtener más información sobre cadenas de acceso, consulte Especificación de permisos mediante una cadena de acceso.

```
import boto3
import logging

logging.basicConfig(level=logging.INFO)
client = boto3.client('elasticache')

def check_user_exists(UserId):
    """Checks if UserId exists

    Returns True if UserId exists, otherwise False
    :param UserId: ElastiCache User ID
    :return: True|False
    """
    try:
        response = client.describe_users(
            UserId=UserId,
        )
```

```
        if response['Users'][0]['UserId'].lower() == UserId.lower():
            return True
    except Exception as e:
        if e.response['Error']['Code'] == 'UserNotFound':
            logging.info(e.response['Error'])
            return False
        else:
            raise

def check_group_exists(UserGroupId):
    """Checks if UserGroupID exists

    Returns True if Group ID exists, otherwise False
    :param UserGroupId: ElastiCache User ID
    :return: True|False
    """

    try:
        response = client.describe_user_groups(
            UserGroupId=UserGroupId
        )
        if response['UserGroups'][0]['UserGroupId'].lower() == UserGroupId.lower():
            return True
    except Exception as e:
        if e.response['Error']['Code'] == 'UserGroupNotFound':
            logging.info(e.response['Error'])
            return False
        else:
            raise

def create_user(UserId=None,UserName=None,Password=None,AccessString=None):
    """Creates a new user

    Returns the ARN for the newly created user or the error message
    :param UserId: ElastiCache user ID. User IDs must be unique
    :param UserName: ElastiCache user name. ElastiCache allows multiple users with the
 same name as long as the associated user ID is unique.
    :param Password: Password for user. Must have at least 16 chars.
    :param AccessString: Access string with the permissions for the user.
    :return: user ARN
    """
    try:
        response = client.create_user(
            UserId=UserId,
```

```
            UserName=UserName,
            Engine='Redis',
            Passwords=[Password],
            AccessString=AccessString,
            NoPasswordRequired=False
        )
        return response['ARN']
    except Exception as e:
        logging.info(e.response['Error'])
        return e.response['Error']

def create_group(UserGroupId=None, UserIds=None):
    """Creates a new group.
    A default user is required (mandatory) and should be specified in the UserIds list

    Return: Group ARN
    :param UserIds: List with user IDs to be associated with the new group. A default
 user is required
    :param UserGroupId: The ID (name) for the group
    :return: Group ARN
    """
    try:
        response = client.create_user_group(
            UserGroupId=UserGroupId,
            Engine='Redis',
            UserIds=UserIds
        )
        return response['ARN']
    except Exception as e:
        logging.info(e.response['Error'])


if __name__ == '__main__':

    groupName='mygroup2'
    userName = 'myuser2'
    userId=groupName+'-'+userName

    # Creates a new user if the user ID does not exist.
    for tmpUserId,tmpUserName in [ (userId,userName), (groupName+'-
default','default')]:
        if not check_user_exists(tmpUserId):
            response=create_user(UserId=tmpUserId,
 UserName=EXAMPLE,Password=EXAMPLE,AccessString='on ~* +@all')
```

```
            logging.info(response)
        # assigns the new user ID to the user group
    if not check_group_exists(groupName):
        UserIds = [ userId , groupName+'-default']
        response=create_group(UserGroupId=groupName,UserIds=UserIds)
        logging.info(response)
```

Para ejecutar el programa, introduzca el siguiente comando:

```
python UserAndUserGroups.py
```

Tutorial: conexión a ElastiCache

En los siguientes ejemplos, se utiliza el cliente de Valkey o Redis OSS para conectarse a ElastiCache.

Temas

- Conexión a un clúster en modo de clúster deshabilitado
- Conexión a un clúster en modo de clúster habilitado

Conexión a un clúster en modo de clúster deshabilitado

Copie el siguiente programa y péguelo en un archivo denominado ConnectClusterModeDisabled.py. Actualice el mecanismo de suministro de credenciales. Las credenciales de este ejemplo se muestran como reemplazables y se les asigna un elemento no declarado. Evite codificar las credenciales de forma rígida.

```
from redis import Redis
import logging

logging.basicConfig(level=logging.INFO)
redis = Redis(host='primary.xxx.yyyyyy.zzz1.cache.amazonaws.com', port=6379,
 decode_responses=True, ssl=True, username=example, password=EXAMPLE)

if redis.ping():
    logging.info("Connected to Redis")
```

Para ejecutar el programa, introduzca el siguiente comando:

```
python ConnectClusterModeDisabled.py
```

Conexión a un clúster en modo de clúster habilitado

Copie el siguiente programa y péguelo en un archivo denominado ConnectClusterModeEnabled.py.

```
from rediscluster import RedisCluster
import logging

logging.basicConfig(level=logging.INFO)
redis = RedisCluster(startup_nodes=[{"host":
 "xxx.yyy.clustercfg.zzz1.cache.amazonaws.com","port": "6379"}],
 decode_responses=True,skip_full_coverage_check=True)

if redis.ping():
    logging.info("Connected to Redis")
```

Para ejecutar el programa, introduzca el siguiente comando:

```
python ConnectClusterModeEnabled.py
```

Ejemplos de uso

En los siguientes ejemplos, se utiliza el SDK de boto3 para que ElastiCache funcione con ElastiCache para Redis OSS.

Temas

- Establecer y obtener cadenas
- Establecer y obtener un hash con varios elementos
- Publicar (escribir) y suscribirse (leer) desde un canal Pub/Sub
- Escribir y leer a partir de un flujo

Establecer y obtener cadenas

Copie el siguiente programa y péguelo en un archivo denominado SetAndGetStrings.py.

```
import time
import logging
logging.basicConfig(level=logging.INFO,format='%(asctime)s: %(message)s')

keyName='mykey'
currTime=time.ctime(time.time())
```

```
# Set the key 'mykey' with the current date and time as value.
# The Key will expire and removed from cache in 60 seconds.
redis.set(keyName, currTime, ex=60)

# Sleep just for better illustration of TTL (expiration) value
time.sleep(5)

# Retrieve the key value and current TTL
keyValue=redis.get(keyName)
keyTTL=redis.ttl(keyName)

logging.info("Key {} was set at {} and has {} seconds until expired".format(keyName,
 keyValue, keyTTL))
```

Para ejecutar el programa, introduzca el siguiente comando:

```
python SetAndGetStrings.py
```

Establecer y obtener un hash con varios elementos

Copie el siguiente programa y péguelo en un archivo denominado SetAndGetHash.py.

```
import logging
import time

logging.basicConfig(level=logging.INFO,format='%(asctime)s: %(message)s')

keyName='mykey'
keyValues={'datetime': time.ctime(time.time()), 'epochtime': time.time()}

# Set the hash 'mykey' with the current date and time in human readable format
 (datetime field) and epoch number (epochtime field).
redis.hset(keyName, mapping=keyValues)

# Set the key to expire and removed from cache in 60 seconds.
redis.expire(keyName, 60)

# Sleep just for better illustration of TTL (expiration) value
time.sleep(5)

# Retrieves all the fields and current TTL
keyValues=redis.hgetall(keyName)
keyTTL=redis.ttl(keyName)
```

```
logging.info("Key {} was set at {} and has {} seconds until expired".format(keyName,
 keyValues, keyTTL))
```

Para ejecutar el programa, introduzca el siguiente comando:

```
python SetAndGetHash.py
```

Publicar (escribir) y suscribirse (leer) desde un canal Pub/Sub

Copie el siguiente programa y péguelo en un archivo denominado PubAndSub.py.

```
import logging
import time

def handlerFunction(message):
    """Prints message got from PubSub channel to the log output

    Return None
    :param message: message to log
    """
    logging.info(message)

logging.basicConfig(level=logging.INFO)
redis = Redis(host="redis202104053.tihewd.ng.0001.use1.cache.amazonaws.com", port=6379,
 decode_responses=True)


# Creates the subscriber connection on "mychannel"
subscriber = redis.pubsub()
subscriber.subscribe(**{'mychannel': handlerFunction})

# Creates a new thread to watch for messages while the main process continues with its
 routines
thread = subscriber.run_in_thread(sleep_time=0.01)

# Creates publisher connection on "mychannel"
redis.publish('mychannel', 'My message')

# Publishes several messages. Subscriber thread will read and print on log.
while True:
    redis.publish('mychannel',time.ctime(time.time()))
    time.sleep(1)
```

Para ejecutar el programa, introduzca el siguiente comando:

```
python PubAndSub.py
```

Escribir y leer a partir de un flujo

Copie el siguiente programa y péguelo en un archivo denominado ReadWriteStream.py.

```
from redis import Redis
import redis.exceptions as exceptions
import logging
import time
import threading

logging.basicConfig(level=logging.INFO)

def writeMessage(streamName):
    """Starts a loop writting the current time and thread name to 'streamName'

    :param streamName: Stream (key) name to write messages.
    """
    fieldsDict={'writerId':threading.currentThread().getName(),'myvalue':None}
    while True:
        fieldsDict['myvalue'] = time.ctime(time.time())
        redis.xadd(streamName,fieldsDict)
        time.sleep(1)

def readMessage(groupName=None,streamName=None):
    """Starts a loop reading from 'streamName'
    Multiple threads will read from the same stream consumer group. Consumer group is
 used to coordinate data distribution.
    Once a thread acknowleges the message, it won't be provided again. If message
 wasn't acknowledged, it can be served to another thread.

    :param groupName: stream group were multiple threads will read.
    :param streamName: Stream (key) name where messages will be read.
    """

    readerID=threading.currentThread().getName()
    while True:
        try:
            # Check if the stream has any message
            if redis.xlen(streamName)>0:
```

```
                # Check if if the messages are new (not acknowledged) or not (already
 processed)
                streamData=redis.xreadgroup(groupName,readerID,
{streamName:'>'},count=1)
                if len(streamData) > 0:
                    msgId,message = streamData[0][1][0]
                    logging.info("{}: Got {} from ID
 {}".format(readerID,message,msgId))
                    #Do some processing here. If the message has been processed
 sucessfuly, acknowledge it and (optional) delete the message.
                    redis.xack(streamName,groupName,msgId)
                    logging.info("Stream message ID {} read and processed successfuly
 by {}".format(msgId,readerID))
                    redis.xdel(streamName,msgId)
            else:
                pass
        except:
            raise

        time.sleep(0.5)

# Creates the stream 'mystream' and consumer group 'myworkergroup' where multiple
 threads will write/read.
try:
    redis.xgroup_create('mystream','myworkergroup',mkstream=True)
except exceptions.ResponseError as e:
    logging.info("Consumer group already exists. Will continue despite the error:
 {}".format(e))
except:
    raise

# Starts 5 writer threads.
for writer_no in range(5):
    writerThread = threading.Thread(target=writeMessage, name='writer-'+str(writer_no),
 args=('mystream',),daemon=True)
    writerThread.start()

# Starts 10 reader threads
for reader_no in range(10):
    readerThread = threading.Thread(target=readMessage, name='reader-'+str(reader_no),
 args=('myworkergroup','mystream',),daemon=True)
    readerThread.daemon = True
    readerThread.start()
```

```
# Keep the code running for 30 seconds
time.sleep(30)
```

Para ejecutar el programa, introduzca el siguiente comando:

```
python ReadWriteStream.py
```

Tutorial: configuración de Lambda para acceder a ElastiCache en una VPC

En este tutorial, puede aprender a crear una caché sin servidor de ElastiCache, a crear una función de Lambda, a probar la función de Lambda y, si lo desea, a limpiarla posteriormente.

Temas

- Paso 1: creación de una caché sin servidor de ElastiCache.
- Paso 2: creación de una función de Lambda para ElastiCache
- Paso 3: prueba de la función de Lambda con ElastiCache
- Paso 4: limpieza (opcional)

Paso 1: creación de una caché sin servidor de ElastiCache.

Siga estos pasos para crear una caché sin servidor.

Paso 1.1: creación de una caché sin servidor

En este paso, creará una caché sin servidor en la VPC predeterminada de Amazon en la región us-east-1 de su cuenta mediante la AWS Command Line Interface (CLI). Para obtener más información sobre la creación de la caché sin servidor mediante la consola o la API de ElastiCache, consulte Creación de una caché sin servidor de Redis OSS.

```
aws elasticache create-serverless-cache \
  --serverless-cache-name cache-01  \
--description "ElastiCache IAM auth application" \
--engine valkey
```

Como puede ver, el valor del campo Estado es `CREATING`. ElastiCache puede tardar un minuto en terminar de crear su caché.

Paso 1.2: copia del punto de conexión de caché sin servidor

Verifique que ElastiCache para Redis OSS haya terminado de crear la caché con el comando `describe-serverless-caches`.

```
aws elasticache describe-serverless-caches \
--serverless-cache-name cache-01
```

Copie la dirección del punto de conexión que aparece en el resultado. Necesitará esta dirección cuando cree el paquete de implementación para la función de Lambda.

Paso 1.3: creación de un rol de IAM

1. Cree un documento de política de confianza de IAM, como se muestra a continuación, para el rol que permita a la cuenta asumir el nuevo rol. Guarde la política en un archivo denominado trust-policy.json.

   JSON

   ```
   {
   "Version":"2012-10-17",
       "Statement": [{
           "Effect": "Allow",
           "Action": "sts:AssumeRole",
           "Resource": "arn:aws:iam::*:role/*"
       },
       {
         "Effect": "Allow",
         "Action": "sts:AssumeRole",
         "Resource": "arn:aws:iam::*:role/*"
       }]
   }
   ```

2. Cree un documento de política de IAM, como se muestra a continuación. Guarde la política en un archivo denominado policy.json.

   JSON

   ```
   {
   ```

```
"Version":"2012-10-17",
  "Statement": [
    {
"Effect" : "Allow",
      "Action" : [
        "elasticache:Connect"
      ],
      "Resource" : [
        "arn:aws:elasticache:us-
east-1:123456789012:serverlesscache:cache-01",
        "arn:aws:elasticache:us-east-1:123456789012:user:iam-user-01"
      ]
    }
  ]
}
```

3. Crear un rol de IAM.

```
aws iam create-role \
--role-name "elasticache-iam-auth-app" \
--assume-role-policy-document file://trust-policy.json
```

4. Creación de la política de IAM.

```
aws iam create-policy \
  --policy-name "elasticache-allow-all" \
  --policy-document file://policy.json
```

5. Adjunte la política de IAM al rol.

```
aws iam attach-role-policy \
 --role-name "elasticache-iam-auth-app" \
 --policy-arn "arn:aws:iam::123456789012:policy/elasticache-allow-all"
```

Paso 1.4: creación de un usuario predeterminado

1. Cree un nuevo usuario predeterminado.

```
aws elasticache create-user \
  --user-name default \
--user-id default-user-disabled \
```

```
--engine redis \
--authentication-mode Type=no-password-required \
--access-string "off +get ~keys*"
```

2. Cree un nuevo usuario habilitado para IAM.

```
aws elasticache create-user \
  --user-name iam-user-01 \
--user-id iam-user-01 \
--authentication-mode Type=iam \
--engine redis \
--access-string "on ~* +@all"
```

3. Cree un grupo de usuarios y asocie al usuario.

```
aws elasticache create-user-group \
  --user-group-id iam-user-group-01 \
--engine redis \
--user-ids default-user-disabled iam-user-01

aws elasticache modify-serverless-cache \
  --serverless-cache-name cache-01  \
--user-group-id iam-user-group-01
```

Paso 2: creación de una función de Lambda para ElastiCache

Para crear una función de Lambda para acceder a la caché de ElastiCache, siga estos pasos.

Paso 2.1: creación de una función de Lambda

En este tutorial, proporcionamos código de ejemplo en Python para su función de Lambda.

Python

El siguiente ejemplo de código de Python lee y escribe un elemento en la caché de ElastiCache. Copie el código y guárdelo en un archivo con el nombre `app.py`. Asegúrese de reemplazar el valor `elasticache_endpoint` del código por la dirección del punto de conexión que copió en el paso anterior.

```
from typing import Tuple, Union
from urllib.parse import ParseResult, urlencode, urlunparse
```

```
import botocore.session
import redis
from botocore.model import ServiceId
from botocore.signers import RequestSigner
from cachetools import TTLCache, cached
import uuid

class ElastiCacheIAMProvider(redis.CredentialProvider):
    def __init__(self, user, cache_name, is_serverless=False, region="us-east-1"):
        self.user = user
        self.cache_name = cache_name
        self.is_serverless = is_serverless
        self.region = region

        session = botocore.session.get_session()
        self.request_signer = RequestSigner(
            ServiceId("elasticache"),
            self.region,
            "elasticache",
            "v4",
            session.get_credentials(),
            session.get_component("event_emitter"),
        )

    # Generated IAM tokens are valid for 15 minutes
    @cached(cache=TTLCache(maxsize=128, ttl=900))
    def get_credentials(self) -> Union[Tuple[str], Tuple[str, str]]:
        query_params = {"Action": "connect", "User": self.user}
        if self.is_serverless:
            query_params["ResourceType"] = "ServerlessCache"
        url = urlunparse(
            ParseResult(
                scheme="https",
                netloc=self.cache_name,
                path="/",
                query=urlencode(query_params),
                params="",
                fragment="",
            )
        )
        signed_url = self.request_signer.generate_presigned_url(
            {"method": "GET", "url": url, "body": {}, "headers": {}, "context": {}},
            operation_name="connect",
            expires_in=900,
```

```
            region_name=self.region,
        )
        # RequestSigner only seems to work if the URL has a protocol, but
        # Elasticache only accepts the URL without a protocol
        # So strip it off the signed URL before returning
        return (self.user, signed_url.removeprefix("https://"))

def lambda_handler(event, context):
    username = "iam-user-01" # replace with your user id
    cache_name = "cache-01" # replace with your cache name
    elasticache_endpoint = "cache-01-xxxxx.serverless.use1.cache.amazonaws.com" #
 replace with your cache endpoint
    creds_provider = ElastiCacheIAMProvider(user=username, cache_name=cache_name,
 is_serverless=True)
    redis_client = redis.Redis(host=elasticache_endpoint, port=6379,
 credential_provider=creds_provider, ssl=True, ssl_cert_reqs="none")

    key='uuid'
    # create a random UUID - this will be the sample element we add to the cache
    uuid_in = uuid.uuid4().hex
    redis_client.set(key, uuid_in)
    result = redis_client.get(key)
    decoded_result = result.decode("utf-8")
    # check the retrieved item matches the item added to the cache and print
    # the results
    if decoded_result == uuid_in:
        print(f"Success: Inserted {uuid_in}. Fetched {decoded_result} from Valkey.")
    else:
        raise Exception(f"Bad value retrieved. Expected {uuid_in}, got
 {decoded_result}")

    return "Fetched value from Valkey"
```

Este código usa la biblioteca redis-py de Python para colocar elementos en la caché y recuperarlos. Este código usa cachetools para almacenar en caché durante 15 minutos los tokens de autenticación de IAM generados. Para crear un paquete de implementación que contenga redis-py y cachetools, siga estos pasos.

En el directorio del proyecto que contiene el archivo de código fuente app.py, cree un paquete de carpetas en el que instalar las bibliotecas redis-py y cachetools.

```
mkdir package
```

Instale redis-py y cachetools usando pip.

```
pip install --target ./package redis
pip install --target ./package cachetools
```

Cree un archivo .zip que contenga las bibliotecas redis-py y cachetools. En Linux y macOS, ejecute el siguiente comando. En Windows, cree un archivo .zip con las bibliotecas redis-py y cachetools en la raíz con la utilidad de compresión zip que prefiera.

```
cd package
zip -r ../my_deployment_package.zip .
```

Añada el código de función al archivo .zip. En Linux y macOS, ejecute el siguiente comando. En Windows, añada app.py al directorio raíz del archivo .zip con la utilidad de compresión zip que prefiera.

```
cd ..
zip my_deployment_package.zip app.py
```

Paso 2.2: crear el rol de IAM (rol de ejecución)

Asocie la política administrada de AWS denominada `AWSLambdaVPCAccessExecutionRole` al rol.

```
aws iam attach-role-policy \
 --role-name "elasticache-iam-auth-app" \
 --policy-arn "arn:aws:iam::aws:policy/service-role/AWSLambdaVPCAccessExecutionRole"
```

Paso 2.3: cargar el paquete de despliegue (crear la función de Lambda)

En este paso, se crea la función de Lambda (AccessValkey) utilizando el comando create-function de la AWS CLI.

Desde el directorio del proyecto que contiene el archivo .zip del paquete de implementación, ejecute el siguiente comando `create-function` de la CLI de Lambda.

Para la opción del rol, utilice el ARN del rol de ejecución que creó en el paso anterior. Para vpc-config, introduzca listas separadas por comas de las subredes de la VPC predeterminada y el ID del grupo de seguridad de la VPC predeterminada. Puede encontrar estos valores en la Consola

de Amazon VPC. Para buscar las subredes de la VPC predeterminada, seleccione Sus VPC y, a continuación, seleccione la VPC predeterminada de su cuenta de AWS. Para buscar el grupo de seguridad de esta VPC, vaya a Seguridad y elija Grupos de seguridad. Compruebe que ha seleccionado la región us-east-1.

```
aws lambda create-function \
--function-name AccessValkey  \
--region us-east-1 \
--zip-file fileb://my_deployment_package.zip \
--role arn:aws:iam::123456789012:role/elasticache-iam-auth-app \
--handler app.lambda_handler \
--runtime python3.12  \
--timeout 30 \
--vpc-config SubnetIds=comma-separated-vpc-subnet-ids,SecurityGroupIds=default-
security-group-id
```

Paso 3: prueba de la función de Lambda con ElastiCache

En este paso, invocará la función de Lambda manualmente utilizando el comando de invocación. Cuando la función de Lambda se ejecuta, genera un UUID y lo escribe en la caché de ElastiCache especificada en el código de Lambda. A continuación, la función de Lambda recupera el elemento de la caché.

1. Invoque la función de Lambda (AccessValkey) mediante el comando de invocación AWS Lambda.

   ```
   aws lambda invoke \
   --function-name AccessValkey  \
   --region us-east-1 \
   output.txt
   ```

2. Compruebe que la función de Lambda se ha ejecutado correctamente del modo siguiente:

   - Revise el archivo output.txt.
   - Para comprobar los resultados en Registros de CloudWatch, abra la consola de CloudWatch y elija el grupo de registros para su función (/aws/lambda/AccessValkey). El flujo de registro debería contener una salida similar a lo siguiente:

     ```
     Success: Inserted 826e70c5f4d2478c8c18027125a3e01e. Fetched
      826e70c5f4d2478c8c18027125a3e01e from Valkey.
     ```

- Revise los resultados en la consola de AWS Lambda.

Paso 4: limpieza (opcional)

Para limpiar, siga estos pasos.

Paso 4.1: eliminación de la función de Lambda

```
aws lambda delete-function \
 --function-name AccessValkey
```

Paso 4.2: eliminación de la caché sin servidor

Elimine la caché.

```
aws elasticache delete-serverless-cache \
 --serverless-cache-name cache-01
```

Quite usuarios y grupos de usuarios.

```
aws elasticache delete-user \
 --user-id default-user-disabled

aws elasticache delete-user \
 --user-id iam-user-01

aws elasticache delete-user-group \
 --user-group-id iam-user-group-01
```

Paso 4.3: eliminación de políticas y el rol de IAM

```
aws iam detach-role-policy \
 --role-name "elasticache-iam-auth-app" \
 --policy-arn "arn:aws:iam::123456789012:policy/elasticache-allow-all"

aws iam detach-role-policy \
--role-name "elasticache-iam-auth-app" \
--policy-arn "arn:aws:iam::aws:policy/service-role/AWSLambdaVPCAccessExecutionRole"

aws iam delete-role \
```

```
--role-name "elasticache-iam-auth-app"

aws iam delete-policy \
 --policy-arn "arn:aws:iam::123456789012:policy/elasticache-allow-all"
```

Creación y administración de un clúster de ElastiCache basado en nodos

Si necesita un control detallado del clúster de ElastiCache, puede crear un clúster basado en nodos. ElastiCache le permite operar un clúster basado en nodos eligiendo el tipo de nodo, la cantidad de nodos y la ubicación de los nodos en las zonas de disponibilidad de AWS para el clúster. Como ElastiCache es un servicio totalmente administrado, gestiona automáticamente el aprovisionamiento de hardware, la supervisión, las sustituciones de nodos y la aplicación de parches de software para el clúster.

Para obtener información sobre la configuración, consulte Configuración de ElastiCache. Para obtener más información sobre la administración, la actualización o la eliminación de nodos o clústeres, consulte Administración de nodos en ElastiCache. Si desea obtener información general sobre los componentes principales de una implementación de Amazon ElastiCache cuando cree su propio clúster de ElastiCache basado en nodos, consulte estos conceptos básicos.

Temas

- Componentes y características de ElastiCache
- Terminología de ElastiCache
- Tutorial: Cómo crear un clúster de ElastiCache basado en nodos
- Eliminación de un clúster
- Otros tutoriales y vídeos de ElastiCache
- Administración de nodos en ElastiCache
- Administración de clústeres en ElastiCache
- Comparación de los clústeres Valkey, Memcached y Redis OSS basados en nodos
- Migración en línea para Valkey o Redis OSS
- Elección de regiones y zonas de disponibilidad para ElastiCache

Componentes y características de ElastiCache

A continuación, encontrará información general sobre los componentes principales de una implementación de Amazon ElastiCache.

Temas

- Nodos de ElastiCache
- Particiones de ElastiCache
- Clústeres de ElastiCache
- Replicación de ElastiCache
- Puntos de conexión de ElastiCache
- Grupos de parámetros de ElastiCache
- Seguridad de ElastiCache
- Grupos de subredes de ElastiCache
- Copias de seguridad de ElastiCache
- Eventos de ElastiCache

Nodos de ElastiCache

Un nodo es el componente básico más pequeño de toda implementación de ElastiCache. Un nodo puede existir de forma aislada o con alguna relación con otros nodos.

Un nodo es una parción de tamaño fijo de RAM segura conectada a la red. Cada nodo ejecuta una instancia del motor y la versión elegidos al crear el clúster. Si es necesario, puede escalar los nodos de un clúster para ampliar o reducir a un tipo de instancia diferente. Para obtener más información, consulte Escalado de ElastiCache.

Todos los nodos contenidos en un clúster son del mismo tipo de instancia y ejecutan el mismo motor de caché. Cada nodo de caché tiene su propio puerto y nombre de servicio de nombres de dominio (DNS). Se admiten varios tipos de nodos de caché, cada uno de los cuales tiene una cantidad diferente de memoria asociada. Para ver una lista de los tipos de instancia de nodos admitidos, consulte Tipos de nodos compatibles.

Puede adquirir nodos a medida que sea necesario, donde solo paga por el uso que haga de un nodo. También puede adquirir nodos reservados a una tarifa por hora más reducida. Si la tasa de uso es alta, adquirir nodos reservados podría ayudar a que ahorre dinero. Supongamos que el clúster está casi siempre en uso, por lo que de vez en cuando agregue nodos para gestionar los picos de uso. En este caso, puede adquirir una serie de nodos reservados para ejecutar la mayor parte del tiempo. Luego, puede adquirir nodos de pago por uso para las veces que necesite agregar nodos. Para obtener más información sobre los nodos reservados, consulte Nodos reservados.

Para obtener más información sobre los nodos, consulte Administración de nodos en ElastiCache.

Particiones de ElastiCache

Una partición de Valkey o Redis OSS (denominada grupo de nodos en la API y en la CLI) es un conjunto de uno a seis nodos relacionados. Un clúster de Valkey o Redis OSS con el modo de clúster habilitado siempre tiene al menos una partición.

Las particiones son un método de división de bases de datos que separa las bases de datos grandes en partes más pequeñas, rápidas y fáciles de administrar denominadas particiones de datos. Esto puede aumentar la eficiencia de la base de datos al distribuir las operaciones en varias secciones independientes. El uso de particiones puede tener muchas ventajas, como la mejora del rendimiento, la escalabilidad y la rentabilidad.

Los clústeres de Valkey y Redis OSS con el modo de clúster habilitado pueden tener hasta 500 particiones, con sus datos particionados en las particiones. El límite de nodos o particiones se puede aumentar a un máximo de 500 por clúster si la versión del motor de Valkey o Redis OSS es 5.0.6 o posteriores. Por ejemplo, puede elegir configurar un clúster de 500 nodos que oscila entre 83 particiones (uno primario y 5 réplicas por partición) y 500 particiones (único primario y sin réplicas). Asegúrese de que hay suficientes direcciones IP disponibles para acomodar el aumento. Algunos problemas comunes incluyen que las subredes del grupo de subredes tienen un rango CIDR demasiado pequeño o que otros clústeres comparten y utilizan considerablemente las subredes. Para obtener más información, consulte Creación de un grupo de subredes. Para las versiones inferiores a 5.0.6, el límite es de 250 por clúster.

Para solicitar un aumento del límite, consulte AWS Service Limits y elija el tipo de límite Nodes per cluster per instance type (Nodos por clúster por tipo de instancias).

Una partición de varios nodos implementa la reproducción al tener un nodo primario de lectura/escritura y 1 a 5 nodos de réplica. Para obtener más información, consulte Alta disponibilidad a través de grupos de reproducción.

Para obtener más información acerca de las particiones, consulte Uso de particiones en ElastiCache.

Clústeres de ElastiCache

Un clúster es una agrupación lógica de uno o más nodos. Los datos se particionan entre los nodos de un clúster de Memcached y entre las particiones de un clúster de Valkey o Redis OSS que tenga habilitado el modo de clúster.

Muchas operaciones de ElastiCache se encuentran destinadas a los clústeres:

- creación de un clúster
- Modificación de un clúster
- Uso de instantáneas de un clúster (todas las versiones de Redis)
- Eliminación de un clúster
- Visualización de elementos de un clúster
- Adición o eliminación de etiquetas de asignación de costos en un clúster

Para obtener información más detallada, consulte los siguientes temas relacionados:

- Administración de clústeres en ElastiCache y Administración de nodos en ElastiCache

  Información acerca de los clústeres, nodos, y operaciones relacionadas.
- AWS Límites de servicio de : Amazon ElastiCache

  Información sobre límites de ElastiCache, como el número máximo de nodos o clústeres. Para superar algunos de estos límites, puede realizar una solicitud mediante el Formulario de solicitud de nodo de caché de Amazon ElastiCache.
- Mitigación de errores

  Información sobre la mejora de la tolerancia a errores de los clústeres y grupos de replicación de Valkey o Redis OSS.

Configuraciones típicas de clúster

A continuación se muestran configuraciones típicas de clúster.

Clústeres de Valkey o Redis OSS

Los clústeres de Valkey o Redis OSS con el modo de clúster deshabilitado siempre contienen solo una partición (en la API y en la CLI, solo un grupo de nodos). Una partición de Valkey o Redis OSS contiene de uno a seis nodos. Si hay más de un nodo en una partición, esta admite la reproducción. En este caso, un nodo es el nodo primario de lectura/escritura y los demás son nodos de réplica de solo lectura.

Para mejorar la tolerancia a errores, le recomendamos tener al menos dos nodos en un clúster de Valkey o Redis OSS y habilitar multi-AZ. Para obtener más información, consulte Mitigación de errores.

A medida que cambie la demanda en su clúster de Valkey o Redis OSS, puede reducir o escalar verticalmente. Para ello, mueva el clúster a un tipo de instancia de nodo diferente. Si la aplicación tiene una gran intensidad de lectura, recomendamos agregar al clúster réplicas de solo lectura. Al hacerlo, puede distribuir las lecturas entre un número más apropiado de nodos.

También puede utilizar la organización de datos en niveles. Los datos a los que se accede con más frecuencia se almacenan en la memoria y los datos a los que se accede con menos frecuencia se almacenan en el disco. La ventaja de utilizar la organización de datos en niveles es que disminuye las necesidades de memoria. Para obtener más información, consulte Organización de datos en niveles en ElastiCache.

ElastiCache admite cambiar un tipo de nodo de clúster de Valkey o Redis OSS a un tipo de nodo más grande de forma dinámica. Para obtener información acerca del escalado mediante ampliación o reducción, consulte Escalado de clústeres para Valkey o Redis OSS (modo de clúster deshabilitado) o Escalado de nodos de réplica para Valkey o Redis OSS (modo de clúster deshabilitado).

Configuraciones típicas de clúster para Memcached

Memcached admite hasta 300 nodos por cliente y región de AWS, donde cada clúster tiene de 1 a 60 nodos. Los datos se particionan entre los nodos del clúster de Memcached.

Cuando ejecuta el motor de Memcached, los clústeres pueden contener de 1 a 60 nodos. La base de datos se particiona entre los nodos. Su aplicación lee y escribe en cada punto de enlace del nodo. Para obtener más información, consulte Detección automática.

Para mejorar la tolerancia a errores, ubique sus nodos de Memcached en varias zonas de disponibilidad (AZ) en la región de AWS del clúster. De ese modo, un error de una zona de disponibilidad tendrá un impacto mínimo en todo su clúster y su aplicación. Para obtener más información, consulte Mitigación de errores.

A medida que cambie la demanda de su clúster de Memcached, puede escalar de forma ascendente o descendente al agregar o quitar nodos, con lo que se vuelven a particionar los datos entre el nuevo número de nodos. Al particionar los datos, recomendamos hacer un uso consistente de la función hash. Para obtener más información acerca del uso consistente de la función hash, consulte Configuración de su cliente de ElastiCache para un equilibrio de carga eficaz (Memcached).

Replicación de ElastiCache

La replicación de Valkey y Redis OSS se implementa al agrupar de dos a seis nodos en una partición (denominada grupo de nodos en la API y la CLI). Uno de estos nodos es el nodo principal de lectura/

escritura. Todos los demás nodos son nodos de réplicas de solo lectura. Las réplicas solo están disponibles para ElastiCache para Valkey y Redis OSS, y no para ElastiCache para Memcached.

Cada nodo de réplica de lectura mantiene una copia de los datos del nodo principal. Los nodos de réplica usan mecanismos de reproducción asíncronos para mantener la sincronización con el nodo principal. Las aplicaciones pueden leer desde cualquiera de los nodos del clúster, pero solo pueden escribir en los nodos principales. Las réplicas de lectura mejoran la escalabilidad al distribuir las lecturas entre varios puntos de enlace. Las réplicas de lectura también mejoran la tolerancia a errores al mantener varias copias de los datos. Ubicar réplicas de lectura en múltiples zonas de disponibilidad mejora aún más la tolerancia a errores. Para obtener más información sobre la tolerancia a errores, consulte Mitigación de errores.

Los clústeres de Valkey o Redis OSS admiten una sola partición (denominada grupo de nodos en la API y la CLI).

La reproducción desde la perspectiva de la API y la CLI utiliza una terminología diferente para mantener la compatibilidad con las versiones anteriores, pero los resultados son los mismos. En la siguiente tabla se muestran los términos de la API y la CLI para la implementación de la reproducción.

Comparación de la replicación: Valkey o Redis OSS (modo de clúster deshabilitado) y Valkey o Redis OSS (modo de clúster habilitado): el clúster de Valkey o Redis OSS con el modo de clúster habilitado comparado con el clúster de Valkey o Redis OSS con el modo de clúster deshabilitado

En la siguiente tabla, puede encontrar una comparación de las características de los grupos de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) y Valkey o Redis OSS (modo de clúster habilitado).

	Clúster de Valkey o Redis OSS con el modo de clúster deshabilitado	Clúster de Valkey o Redis OSS con el modo de clúster habilitado
Particiones (grupos de nodos)	1	1 a 500
Réplicas para cada partición (grupo de nodos)	0 a 5	0 a 5
Particiones de datos	No	Sí
Agregar o eliminar réplicas	Sí	Sí

	Clúster de Valkey o Redis OSS con el modo de clúster deshabilitado	Clúster de Valkey o Redis OSS con el modo de clúster habilitado
Agregar o eliminar grupos de nodos	No	Sí
Compatible con el escalado vertical	Sí	Sí
Compatible con las actualizaciones del motor	Sí	Sí
Promociona la réplica a principal	Sí	Automático
Multi-AZ	Opcional	Obligatorio
Copia de seguridad y restauración	Sí	Sí
Notas:		
Si cualquier primario no tiene réplicas y falla, se perderán todos los datos de ese primario.		
Se puede utilizar la copia de seguridad y la restauración para migrar a Valkey o Redis OSS (modo de clúster habilitado).		
Se puede utilizar la copia de seguridad y la restauración para cambiar el tamaño del clúster de Valkey o Redis OSS (modo de clúster habilitado).		

Todas las particiones (grupos de nodos en la API y en la CLI) y todos los nodos deben residir en la misma región de AWS. Sin embargo, puede aprovisionar los nodos individuales en diferentes zonas de disponibilidad dentro de esa región de AWS.

Las réplicas de lectura evitan las posibles pérdidas de datos, ya que los datos se replican a través de dos o varios nodos: el principal y una o varias réplicas de lectura. Para una mayor fiabilidad y una recuperación más rápida, recomendamos crear una o varias réplicas de lectura en distintas zonas de disponibilidad.

También puede aprovechar los almacenes de datos Globales. Con la característica del almacén de datos global de Redis OSS, puede trabajar con replicación completamente administrada, rápida, fiable y segura en todas las regiones de AWS. Con esta característica, puede crear clústeres de réplicas de lectura entre regiones para que ElastiCache permita lecturas de baja latencia y recuperación ante desastres en todas las regiones de AWS. Para obtener más información, consulte Replicación en regiones de AWS mediante almacenes de datos globales.

Replicación: límites y exclusiones

- Multi-AZ no se admite en los tipos de nodo T1.

Puntos de conexión de ElastiCache

Un punto de conexión es la dirección única que utiliza la aplicación para conectarse a un nodo o clúster de ElastiCache.

Puntos de conexión de un solo nodo para Valkey o Redis OSS con el modo de clúster deshabilitado

El punto de conexión de un clúster de Valkey o Redis OSS de un solo nodo se utiliza para conectarse al clúster tanto para lecturas como para escrituras.

Puntos de conexión de varios nodos para Valkey o Redis OSS con el modo de clúster deshabilitado

Un clúster de varios nodos de Valkey o Redis OSS con el modo de clúster deshabilitado tiene dos tipos de puntos de conexión. El punto de conexión principal siempre se conecta al nodo principal en el clúster, incluso si el nodo específico de la función principal cambia. Utilice el punto de conexión principal para todas las escrituras en el clúster.

Use el punto de conexión de lector para dividir de manera uniforme las conexiones entrantes al punto de enlace entre todas las réplicas de lectura. Use los puntos de conexión del nodo para las operaciones de lectura (en la API/CLI, estas se denominan puntos de conexión de lectura).

Puntos de conexión de Valkey o Redis OSS (modo de clúster habilitado)

Un clúster de Valkey o Redis OSS con el modo de clúster habilitado tiene un único punto de conexión de configuración. Al conectarse al punto de conexión de configuración, la aplicación puede detectar los puntos de conexión principal y de lectura de cada partición del clúster.

Para obtener más información, consulte Búsqueda de puntos de conexión en ElastiCache.

Puntos de enlace de ElastiCache for Memcached

Cada nodo de un clúster de Memcached tiene su propio punto de enlace. El clúster también tiene un punto de enlace, denominado punto de enlace de configuración. Si habilita la función Auto Discovery y se conecta al punto de enlace de configuración, su aplicación conocerá automáticamente cada punto de enlace de nodo, incluso después de agregar o quitar nodos en el clúster. Para obtener más información, consulte Detección automática.

Para obtener más información, consulte Búsqueda de puntos de conexión en ElastiCache.

Grupos de parámetros de ElastiCache

Los grupos de parámetros de caché son una forma sencilla de administrar la configuración de tiempo de ejecución para el software del motor compatible. Los parámetros se utilizan para controlar el uso de la memoria, las políticas de expulsión, los tamaños de elementos y mucho más. Un grupo de parámetros de ElastiCache es una colección nombrada de parámetros específicos del motor que se pueden aplicar a un clúster. De este modo, se asegura de que todos los nodos de ese clúster estén configurados exactamente de la misma forma.

Para obtener una lista de los parámetros compatibles, sus valores predeterminados y cuáles se pueden modificar, consulte DescribeEngineDefaultParameters (CLI: describe-engine-default-parameters).

Para obtener información más detallada sobre los grupos de parámetros de ElastiCache, consulte Configuración de los parámetros de motor mediante grupos de parámetros de ElastiCache.

Seguridad de ElastiCache

Para mayor seguridad, el acceso al nodo de ElastiCache se encuentra restringido a las aplicaciones que se ejecutan en las instancias de Amazon EC2 permitidas. Puede controlar las instancias de Amazon EC2 que pueden acceder al clúster mediante los grupos de seguridad.

De forma predeterminada, todos los clústeres de ElastiCache nuevos se lanzan en un entorno de Amazon Virtual Private Cloud (Amazon VPC). Puede utilizar grupos de subredes para conceder acceso al clúster desde las instancias de Amazon EC2 que se ejecutan en subredes concretas.

Además de restringir el acceso a los nodos, ElastiCache es compatible con TLS y con el cifrado en el sitio en los nodos que ejecutan las versiones especificadas de ElastiCache. Para obtener más información, consulte los siguientes temas:

- Seguridad de datos en Amazon ElastiCache
- Autenticación con el comando AUTH de Valkey o Redis OSS

Grupos de subredes de ElastiCache

Un grupo de subredes es una colección de subredes (que suelen ser privadas) que se puede designar para los clústeres que se ejecuten en un entorno de Amazon VPC.

Si crea un clúster en una Amazon VPC, debe especificar un grupo de subredes de caché. ElastiCache utiliza dicho grupo de subredes de caché para elegir una subred y direcciones IP pertenecientes a ella y asociarlas a los nodos de caché.

Para obtener más información sobre el uso de los grupos de subredes de caché en entornos de Amazon VPC, consulte lo siguiente:

- Seguridad de ElastiCache y Amazon VPC
- Paso 3. Autorización del acceso al clúster
- Subredes y grupos de subredes

Copias de seguridad de ElastiCache

Una copia de seguridad es una copia en un momento dado de una caché sin servidor o de un clúster de Valkey o Redis OSS, o de una caché sin servidor de Memcached. Dichas copias de seguridad pueden usarse para restaurar un clúster existente o para propagar datos en un nuevo clúster. Las copias de seguridad se componen de todos los datos en un clúster, además de algunos metadatos.

Dependiendo de la versión de Valkey o Redis OSS que se ejecute en el clúster, el proceso de copia de seguridad requiere diferentes cantidades de memoria reservada para realizarse correctamente. Para obtener más información, consulte los siguientes temas:

- Instantánea y restauración
- Cómo se implementan la sincronización y la copia de seguridad
- Impacto en el rendimiento de las copias de seguridad de los clústeres basados en nodos
- Forma de garantizar que dispone de memoria suficiente para crear una instantánea de Valkey o Redis OSS

Eventos de ElastiCache

Cuando se producen eventos significativos en un clúster, ElastiCache envía una notificación a un tema de Amazon SNS en concreto. Algunos eventos significativos pueden ser, por ejemplo, errores al agregar un nodo, adiciones de nodos correctas, cambios en los grupos de seguridad, entre otros. Al monitorear los eventos clave podrá conocer el estado actual de los clústeres y, según el evento, adoptar medidas correctivas.

Para obtener más información sobre los eventos de ElastiCache consulte Supervisión de eventos de ElastiCache en Amazon SNS.

Terminología de ElastiCache

En octubre de 2016, Amazon ElastiCache lanzó la compatibilidad con Redis OSS 3.2. En ese momento, agregamos compatibilidad con la división de los datos en hasta 500 particiones (denominadas grupos de nodos en la API de ElastiCache y en la AWS CLI). Para conservar la compatibilidad con versiones anteriores, hemos ampliado las operaciones de la versión de la API 2015-02-02 para incluir la nueva funcionalidad de Redis OSS.

Al mismo tiempo, hemos comenzado a utilizar la terminología en la consola de ElastiCache que se utiliza en esta funcionalidad nueva y de forma habitual en el sector. Estos cambios implican que, en algunos aspectos, la terminología usada en la API y la CLI puede ser diferente de la terminología empleada en la consola. La siguiente lista identifica términos que pueden variar entre la API y la CLI y la consola.

Clúster de caché o nodo frente a nodo

Si no hay nodos de réplica, existe una relación de uno a uno entre un nodo y un clúster. Por lo tanto, la consola de ElastiCache solía utilizar los términos indistintamente. A partir de ahora, la consola utiliza el término nodo para todo. La única excepción es el botón Create Cluster, que lanza el proceso para crear un clúster con o sin nodos de réplica.

La AWS CLI y la API de ElastiCache siguen utilizando los términos como antes.

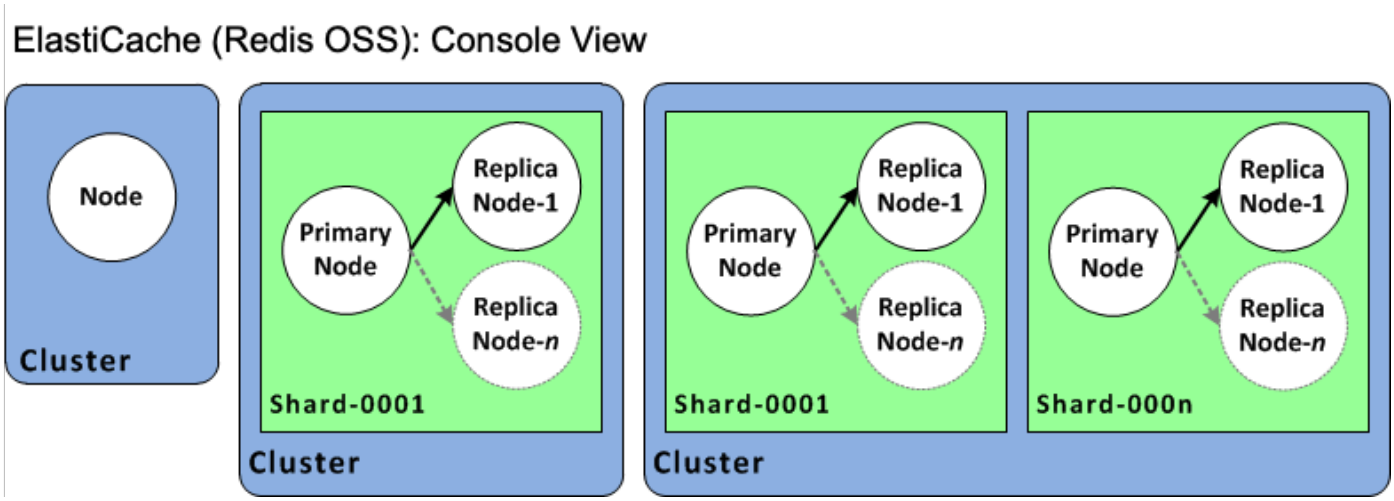
Comparación entre el grupo de replicación de clúster y el grupo de replicación de Valkey o Redis OSS

Ahora, la consola utiliza el término clúster para todos los clústeres de ElastiCache para Redis. La consola utiliza el término clúster en todas estas circunstancias:

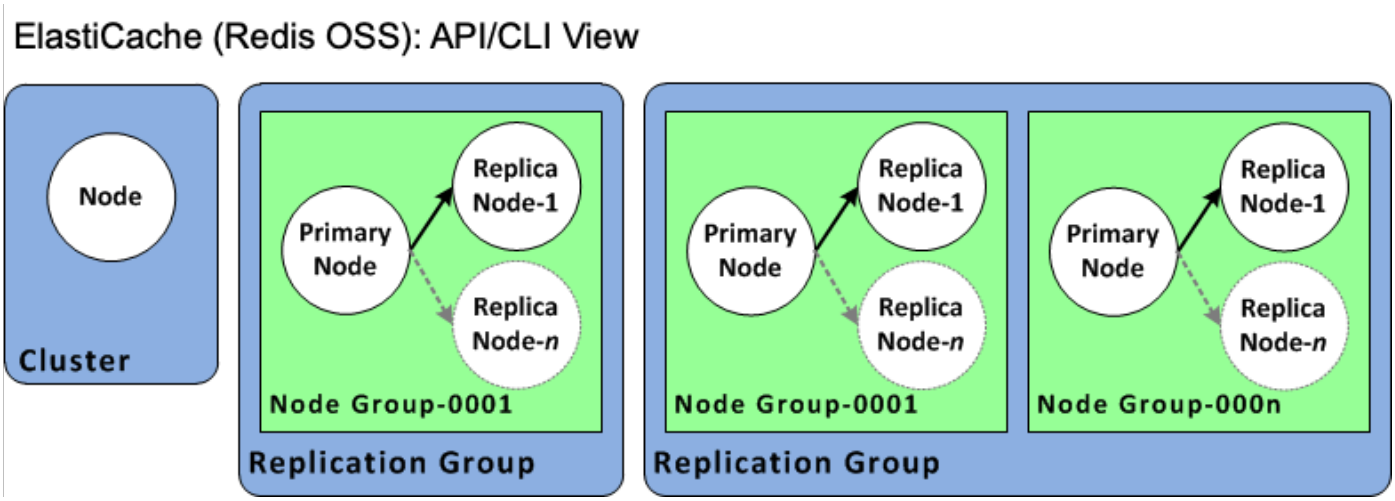
- Cuando el clúster es un clúster de Valkey o Redis OSS de un solo nodo.
- Cuando el clúster es un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) que admite la replicación en una única partición (en la API y en la CLI, denominado grupo de nodos).
- Cuando el clúster es un clúster de Valkey o Redis OSS (modo de clúster habilitado) que admite la replicación de entre 1 a 90 particiones o hasta 500 con una solicitud de aumento de límite. Para solicitar un aumento del límite, consulte AWS Service Limits y elija el tipo de límite Nodes per cluster per instance type (Nodos por clúster por tipo de instancias).

Para obtener más información acerca de los grupos de replicación de Valkey o Redis OSS, consulte. Alta disponibilidad a través de grupos de reproducción

El siguiente diagrama ilustra las distintas topologías de clústeres de ElastiCache para Redis OSS desde la perspectiva de la consola.



Las operaciones de la AWS CLI y la API de ElastiCache siguen distinguiendo los clústeres de ElastiCache para Redis OSS de un solo nodo de los grupos de replicación de varios nodos de Valkey o Redis OSS. El siguiente diagrama ilustra las distintas topologías de ElastiCache para Redis desde la perspectiva de la AWS CLI y la API de ElastiCache.



Grupo de replicación de Valkey o Redis OSS comparado con el almacén de datos global

Un almacén de datos global es una colección de uno o varios clústeres que se replican entre sí entre regiones, mientras que un grupo de replicación de Valkey o Redis OSS replica datos entre un clúster en modo de clúster habilitado con varias particiones. Un almacén de datos global consta de lo siguiente:

- Clúster principal (activo): un clúster principal acepta escrituras que se replican en todos los clústeres dentro del almacén de datos global. Un clúster principal también acepta solicitudes de lectura.
- Clúster secundario (pasivo): un clúster secundario solo acepta solicitudes de lectura y replica las actualizaciones de datos a partir de un clúster principal. Un clúster secundario debe encontrarse en una región de AWS diferente a la del principal.

Para obtener información sobre almacenes de datos globales, consulte Replicación en regiones de AWS mediante los almacenes de datos globales.

Tutorial: Cómo crear un clúster de ElastiCache basado en nodos

A continuación, se explica cómo crear un clúster de ElastiCache basado en nodos para Valkey, Memcached y Redis OSS.

Temas

- Creación de un clúster de ElastiCache basado en nodos para Valkey
- Creación de un clúster de ElastiCache basado en nodos para Redis OSS

Creación de un clúster de ElastiCache basado en nodos para Valkey

A continuación, se indican las medidas puntuales que debe tomar para crear un clúster de ElastiCache basado en nodos para Valkey.

Paso 1: crear un grupo de subredes

Antes de crear el clúster de ElastiCache (Valkey), asegúrese de crear un grupo de subredes. Un grupo de subredes de caché es una colección de subredes que podría designar para sus clústeres en una VPC. Al lanzar un clúster en una VPC, tiene que seleccionar un grupo de subredes de caché. A continuación, ElastiCache utilizará dicho grupo de subredes de caché a fin de asignar direcciones IP de esta subred para cada nodo de caché del clúster.

Cuando cree un nuevo grupo de subredes, tenga en cuenta el número de direcciones IP disponibles. Si la subred tiene pocas direcciones IP libres, el número de nodos que podrá agregar al clúster será limitado. Para solucionar este problema, puede asignar una o varias subredes a un grupo de subredes para, de este modo, disponer de suficientes direcciones IP en la zona de disponibilidad de su clúster. Hecho esto, podrá agregar más nodos a su clúster.

Para obtener más información sobre la configuración de ElastiCache, consulte Configuración de ElastiCache.

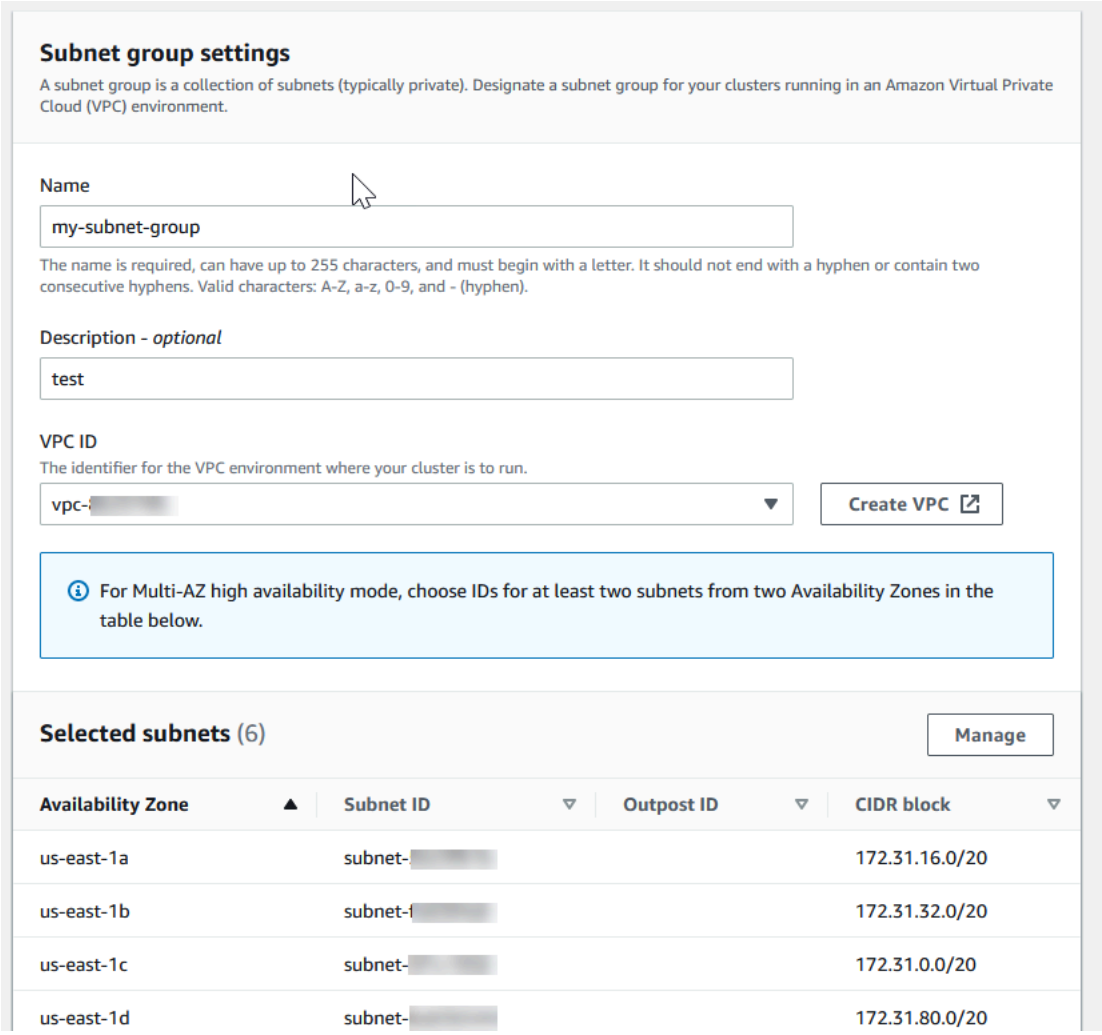
En los siguientes procedimientos, se muestra cómo crear un grupo de subredes denominado `mysubnetgroup` (consola) y la AWS CLI.

Creación de un grupo de subredes (consola)

En el siguiente procedimiento, se muestra cómo crear un grupo de subredes (consola).

Para crear un grupo de subredes (consola)

1. Inicie sesión en la AWS Management Console y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En la lista de navegación, elija Subnet Groups (Grupos de subredes).
3. Elija Create Subnet Group (Crear grupo de subredes).
4. En el asistente Create Subnet Group(Crear grupo de subredes), haga lo siguiente. Una vez que la configuración sea la deseada, elija Yes, Create (Sí, crear).
    a. En el cuadro Name (Nombre), escriba un nombre para el grupo de subredes.
    b. En el cuadro Description(Descripción), escriba la descripción del grupo de subredes.
    c. En el cuadro de ID de la VPC, elija la Amazon VPC que creó anteriormente.
    d. En las listas Zona de disponibilidad y ID de subred, elija la zona de disponibilidad o Uso de Local Zones con ElastiCache y el ID de su subred privada y, a continuación, elija Agregar.



5. En el mensaje de confirmación que aparece, elija Close (Cerrar).

El grupo de subredes nuevo aparecerá en la lista Subnet Groups (Grupos de subredes) de la consola de ElastiCache. En la parte inferior de la ventana, podrá elegir el grupo de subredes para ver detalles tales como todas las subredes asociadas al grupo.

Crear un grupo de subredes (AWS CLI)

En el símbolo del sistema, utilice el comando `create-cache-subnet-group` para crear un grupo de subredes.

Para Linux, macOS o Unix:

```
aws elasticache create-cache-subnet-group \
    --cache-subnet-group-name mysubnetgroup \
    --cache-subnet-group-description "Testing" \
```

```
    --subnet-ids subnet-53df9c3a
```

Para Windows:

```
aws elasticache create-cache-subnet-group ^
    --cache-subnet-group-name mysubnetgroup ^
    --cache-subnet-group-description "Testing" ^
    --subnet-ids subnet-53df9c3a
```

Este comando debería producir un resultado similar al siguiente:

```
{
    "CacheSubnetGroup": {
        "VpcId": "vpc-37c3cd17",
        "CacheSubnetGroupDescription": "Testing",
        "Subnets": [
            {
                "SubnetIdentifier": "subnet-53df9c3a",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2a"
                }
            }
        ],
        "CacheSubnetGroupName": "mysubnetgroup"
    }
}
```

Para obtener más información, consulte el tema AWS CLI de la create-cache-subnet-group.

Paso 2: crear un clúster

Antes de crear un clúster para su uso en producción, obviamente debe considerar cómo configurará el clúster a fin de satisfacer las necesidades del negocio. Estos problemas se abordan en la sección Preparación de un clúster en ElastiCache. A los efectos de este ejercicio de introducción, creará un clúster con el modo de clúster deshabilitado y podrá aceptar los valores de configuración predeterminados donde se apliquen.

El clúster que crea se ejecutará en un entorno real, no en uno de pruebas. Deberá pagar las tarifas de uso estándares de ElastiCache para la instancia hasta que la elimine. Los cargos totales serán mínimos (normalmente menos de un dólar) si completa el ejercicio descrito aquí de una vez y elimina

el clúster al finalizar. Para obtener más información sobre las tarifas de uso de ElastiCache, consulte Amazon ElastiCache.

El clúster se lanza en una nube virtual privada (VPC) en función del servicio de Amazon VPC.

Creación de un clúster de Valkey (modo de clúster deshabilitado) (consola)

Creación de un clúster de Valkey (modo de clúster deshabilitado) con la consola de ElastiCache

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En la lista de la esquina superior derecha, elija la región de AWS en la que desea lanzar este clúster.
3. En el panel de navegación, seleccione Get started (Introducción).
4. Elija Create VPC (Crear una VPC) y siga los pasos que se detallan en Creating a Virtual Private Cloud (VPC) (Creación de una nube privada virtual).
5. En la página del panel de ElastiCache, seleccione Caché de Valkey o Caché de Redis OSS y, a continuación, elija Crear caché de Valkey o Crear caché de Redis OSS.
6. En Cluster settings (Configuración de clúster), realice la siguiente operación:

    a. Elija Configure and create a new cluster (Configurar y crear un nuevo clúster).

    b. Para Cluster mode (Modo clúster), elige Disabled (Desactivado).

    c. Para Cluster info (Información del clúster) introduzca un valor para Name (Nombre).

    d. (Opcional) Introduzca un valor para Description (Descripción).
7. En Location (Ubicación):

    AWS Cloud

    1. Para Nube de AWS, le recomendamos que acepte la configuración predeterminada de Multi-AZ y Conmutación por error automática. Para obtener más información, consulte Minimización del tiempo de inactividad en ElastiCache para Redis OSS con Multi-AZ.
    2. En Cluster settings (Configuración del clúster)

        a. Para Engine version (Versión del motor), elija una versión disponible.

        b. En Port (Puerto), utilice el puerto predeterminado, 6379. Si tiene un motivo para utilizar un puerto diferente, escriba el número de puerto.

c. En Grupo de parámetros, elija un grupo de parámetros o cree uno nuevo. Los grupos de parámetros controlan los parámetros de tiempo de ejecución de su clúster. Para obtener más información acerca de los grupos de parámetros, consulte Parámetros de Valkey y Redis OSS y Creación de un grupo de parámetros de ElastiCache.

> **Note**
>
> Cuando se selecciona un grupo de parámetros para definir los valores de configuración del motor, dicho grupo de parámetros se aplica a todos los clústeres del almacén de datos global. En la página Parameter Groups (Grupos de parámetros), el atributo Global sí/no, indica si un grupo de parámetros forma parte de un almacén de datos global.

d. Para Node type, elija la flecha hacia abajo
(▼).
En el cuadro de diálogo Cambiar tipo de nodo, elija un valor para Familia de instancias para el tipo de nodo que desea. A continuación, elija el tipo de nodo que desea utilizar para este clúster y, a continuación, elija Guardar.

Para obtener más información, consulte Elección del tamaño del nodo.

Si elige un tipo de nodo r6gd, la organización de datos en niveles se activará automáticamente. Para obtener más información, consulte Organización de datos en niveles en ElastiCache.

e. Para Number of replicas (Número de réplicas), elija el número de réplicas de lectura que desee. Si habilitó Multi-AZ, el número debe estar entre 1 y 5.

3. En Connectivity (Conectividad)

a. Para el Network type (Tipo de red), elija las versiones IP que admitirá este clúster.

b. En Subnet groups (Grupos de subredes), elija la subred que desea aplicar a este clúster. ElastiCache utiliza dicho grupo de subredes para elegir una subred y direcciones IP pertenecientes a dicha subred a fin de asociarlas a sus nodos. Los clústeres de ElastiCache requieren una subred de doble pila con direcciones IPv4 e IPv6 asignadas para funcionar en modo de doble pila y una subred solo IPv6 para funcionar como solo IPv6.

Al crear un nuevo grupo de subredes, escriba el VPC ID (ID de VPC) al que pertenece.

Para obtener más información, consulte:

- Elección de un tipo de red en ElastiCache.
- Cree una subred en la VPC.

Si es Uso de Local Zones con ElastiCache, debe crear o elegir una subred que se encuentre en la zona local.

Para obtener más información, consulte Subredes y grupos de subredes.

4. En Availability zone placements Ubicaciones de las zonas de disponibilidad), tiene dos opciones:

   - No preference (Sin preferencia): ElastiCache elige la zona de disponibilidad.
   - Specify availability zones (Especificar zonas de disponibilidad): especifique la zona de disponibilidad para cada clúster.

     Si decide especificar las zonas de disponibilidad, para cada clúster en cada fragmento, elija la zona de disponibilidad de la lista.

   Para obtener más información, consulte Elección de regiones y zonas de disponibilidad para ElastiCache.

5. Elija Next (Siguiente).
6. En Configuración avanzada de Valkey o Redis OSS

   - Para Security (Seguridad):

     i. Para cifrar sus datos, tiene las siguientes opciones:

        - Encryption at rest (Cifrado en reposo): permite el cifrado de los datos almacenados en el disco. Para obtener más información, consulte Cifrado en reposo.

> **Note**
>
> Tiene la opción de suministrar una clave de cifrado distinta al elegir Customer Managed AWS KMS key (Clave KMS de administrada por el cliente) y elegir la clave. Para obtener más información, consulte Uso de claves de AWS KMS administradas por el cliente.

- Encryption in-transit (Cifrado en tránsito): permite el cifrado de datos del cable. Para obtener más información, consulte Cifrado en tránsito. Para Valkey y Redis OSS 6.0 y versiones posteriores, si habilita el cifrado en tránsito, se le pedirá que especifique una de las siguientes opciones de Control de acceso:
  - No Access Control (Sin control de acceso): este es el valor predeterminado. Esto indica que no hay restricciones en el acceso de los usuarios al clúster.
  - User Group Access Control List (Lista de control de acceso del grupo de usuarios): seleccione un grupo de usuarios con un conjunto definido de usuarios que pueden acceder al clúster. Para obtener más información, consulte Administración de grupos de usuarios con la consola y la CLI.
  - Usuario predeterminado de AUTH: un mecanismo de autenticación para el servidor de Redis OSS. Para obtener más información, consulte AUTH.
- AUTH: un mecanismo de autenticación para el servidor de Redis OSS. Para obtener más información, consulte AUTH.

> **Note**
>
> Para Valkey y para Redis OSS desde la versión 3.2.6 en adelante, excepto la versión 3.2.10, AUTH de Redis OSS es la única opción.

ii. En Grupos de seguridad, elija los grupos de seguridad que desea para este clúster. Un grupo de seguridad actúa como un firewall para controlar el acceso de red al clúster. Puede utilizar el grupo de seguridad predeterminado para la VPC o crear uno nuevo.

Para obtener más información sobre los grupos de seguridad, consulte Grupos de seguridad de su VPC en la Guía del usuario de Amazon VPC.

7. En las copias de seguridad automáticas programadas regularmente, seleccione Enable automatic backups (Habilitar copias de seguridad automáticas) y, a continuación, escriba el número de días que desea conservar cada copia de seguridad automática antes de que se elimine automáticamente. Si no quiere tener copias de seguridad automáticas programadas regularmente, desactive la casilla de verificación Enable automatic backups (Habilitar copias de seguridad automáticas). En cualquier caso, siempre tendrá la opción de crear copias de seguridad manuales.

   Para obtener más información acerca de la copia de seguridad y restauración de Redis OSS, consulte Instantánea y restauración.

8. (Opcional) Especifique un periodo de mantenimiento. El periodo de mantenimiento hace alusión al tiempo, generalmente una hora, de cada semana en la que ElastiCache programa el mantenimiento del sistema en su clúster. Puede permitir que ElastiCache elija el día y la hora de su periodo de mantenimiento (No preference [Sin preferencia]), o bien puede elegir el día, la hora y la duración por su cuenta (Specify maintenance window [Especificar periodo de mantenimiento]). Si elige Especificar el periodo de mantenimiento en las listas, seleccione Día de inicio, Hora de inicio y Duración (en horas) para el periodo de mantenimiento. Todas las horas se indican en UCT.

   Para obtener más información, consulte Administración del mantenimiento de clústeres de ElastiCache.

9. (Opcional) Para Logs (Registros):

   - En Log format (Formato de registro), elija Text o JSON.
   - En Destination Type (Tipo de destino), elija CloudWatch Logs o Kinesis Firehose.
   - En Destino de registro, elija Crear nuevo e introduzca el nombre del grupo de registro de Registros de CloudWatch o el nombre del flujo de Firehose, o elija Seleccionar existente y, a continuación, elija el nombre del grupo de registro de Registros de CloudWatch o el nombre del flujo de Firehose.

10. Para Tags (Etiquetas), para ayudarlo a administrar sus clústeres y otros recursos de ElastiCache, puede asignar sus propios metadatos a cada recurso en forma de etiquetas. Para obtener más información, consulte Etiquetado de los recursos de ElastiCache.

11. Elija Siguiente.
12. Revise todas las entradas y opciones y, a continuación, realice todos los cambios necesarios. Cuando haya terminado, elija Create (Crear).

On premises

1. Para On premises (En las instalaciones), le recomendamos que deje Auto-failover (Conmutación por error automática) habilitada. Para obtener más información, consulte Minimización del tiempo de inactividad en ElastiCache para Redis OSS con Multi-AZ.
2. Para terminar de crear el clúster, siga los pasos que se indican en Using Outposts (Uso de Outposts).

En cuanto el estado de su clúster sea available (disponible), podrá conceder a Amazon EC2 acceso a este, así como conectarse a él y comenzar a utilizarlo. Para obtener más información, consulte Paso 3. Autorización del acceso al clúster y Paso 4. Conexión al nodo de un clúster.

> **Important**
>
> Una vez que su clúster esté disponible, se cobrará por cada hora u hora parcial que el clúster esté activo, incluso si no lo está utilizando de forma activa. Para dejar de incurrir en cargos por este clúster, debe eliminarlo. Consulte Eliminación de un clúster en ElastiCache.

Para trabajar con el modo de clúster habilitado, consulte los temas siguientes:

- Para utilizar la consola, consulte Creación de un clúster de Valkey o Redis OSS (modo de clúster habilitado) (consola).
- Para utilizar la AWS CLI, consulte Creación de un clúster de Valkey o Redis OSS (modo de clúster habilitado) (AWS CLI).

Paso 3. Autorización del acceso al clúster

En esta sección se presupone que se encuentra familiarizado con los procesos de lanzar instancias de Amazon EC2 y conectarse a estas. Para obtener más información, consulte la Guía de introducción de Amazon EC2.

Todos los clústeres de ElastiCache se han diseñado para que se pueda acceder a estos desde una instancia de Amazon EC2. Lo más común es obtener acceso a un clúster de ElastiCache desde una instancia de Amazon EC2 en la misma Amazon Virtual Private Cloud (Amazon VPC), este será el caso utilizado para este ejercicio.

De forma predeterminada, el acceso de red al clúster está limitado a la cuenta que se utilizó para crearlo. Antes de poder conectarse a un clúster desde una instancia EC2, debe autorizar a la instancia EC2 el acceso al clúster.

El caso de uso más común es cuando una aplicación implementada en una instancia EC2 debe conectarse a un clúster en la misma VPC. La forma más sencilla de administrar el acceso entre instancias EC2 y clústeres de la misma VPC es realizar lo siguiente:

1. Cree un grupo de seguridad de VPC para su clúster. Este grupo de seguridad se puede utilizar para restringir el acceso a las instancias de clúster. Por ejemplo, puede crear una regla personalizada para este grupo de seguridad que permita el acceso mediante TCP utilizando el puerto que asignó al clúster de base de datos cuando lo creó y una dirección IP que se utilizará para obtener acceso al clúster.

   El puerto predeterminado para los clústeres y grupos de replicación de Valkey o Redis OSS es 6379.

   > **Important**
   >
   > Los grupos de seguridad de Amazon ElastiCache solo son aplicables a los clústeres que no se ejecutan en un entorno de Amazon Virtual Private Cloud (VPC). Si la ejecución tiene lugar en una Amazon Virtual Private Cloud, Security Groups (Grupos de seguridad) no se encontrará disponible en el panel de navegación de la consola.
   > Si ejecuta los nodos de ElastiCache en una Amazon VPC, podrá controlar el acceso a los clústeres con los grupos de seguridad de Amazon VPC, que son distintos de los grupos de seguridad de ElastiCache. Para obtener más información sobre la utilización de ElastiCache en Amazon VPC, consulte Seguridad de ElastiCache y Amazon VPC

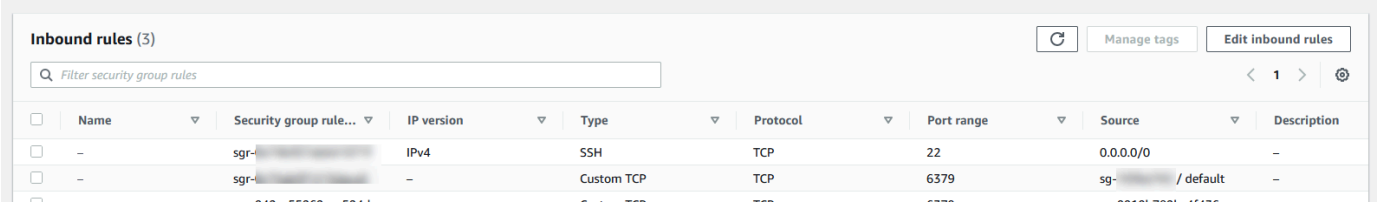
2. Cree un grupo de seguridad de VPC para sus instancias EC2 (servidores web y de aplicaciones). Si es necesario, este grupo de seguridad puede permitir el acceso a la instancia EC2 desde Internet a través de la tabla de enrutamiento de la VPC. Por ejemplo, puede establecer reglas en este grupo de seguridad para permitir el acceso mediante TCP a la instancia EC2 a través del puerto 22.
3. Cree reglas personalizadas en el grupo de seguridad para su clúster que permitan las conexiones desde el grupo de seguridad que creó para las instancias EC2. Esto permitirá a cualquier miembro del grupo de seguridad obtener acceso a los clústeres.

> **Note**
>
> Si planea utilizar Local Zones, asegúrese de que las ha habilitado. Cuando crea un grupo de subredes en esa zona local, la VPC se amplía a dicha zona local y la VPC tratará la subred como cualquier subred de cualquier otra zona de disponibilidad. Todas las gateways y tablas de enrutamiento relevantes se ajustarán de forma automática.

Para crear una regla en un grupo de seguridad de VPC que permita establecer conexiones desde otro grupo de seguridad

1. Inicie sesión en la consola de administración de AWS y abra la consola de Amazon VPC en https://console.aws.amazon.com/vpc.
2. En el panel de navegación, elija Security Groups (Grupos de seguridad).
3. Seleccione o cree un grupo de seguridad que utilizará para sus instancias de clúster. En Inbound Rules (Reglas de entrada), seleccione Edit Inbound Rules (Editar reglas de entrada) y, a continuación, seleccione Add Rule (Agregar regla). Este grupo de seguridad permitirá el acceso a los miembros de otro grupo de seguridad.
4. En Type (Tipo), elija Custom TCP Rule (Personalizar regla de TCP).
   a. En Port Range (Rango de puerto), especifique el puerto que utilizó al crear su clúster.

      El puerto predeterminado para los clústeres y grupos de replicación de Valkey o Redis OSS es 6379.
   b. En el cuadro Source (Fuente), comience a escribir el ID del grupo de seguridad. Desde la lista, seleccione el grupo de seguridad que utilizará para sus instancias de Amazon EC2.
5. Cuando haya terminado, elija Save (Guardar).



Una vez que haya habilitado el acceso, se encontrará listo para conectarse al nodo, como se describe en la siguiente sección.

Para obtener más información sobre cómo obtener acceso al clúster de ElastiCache desde una Amazon VPC diferente, desde una región de AWS diferente o incluso desde la red de la empresa, consulte lo siguiente:

- Patrones de acceso para obtener acceso a una caché de ElastiCache en una Amazon VPC
- Acceso a recursos de ElastiCache desde fuera de AWS

Paso 4. Conexión al nodo de un clúster

Antes de continuar, realice el Paso 3. Autorización del acceso al clúster.

En esta sección se da por sentado que ha creado una instancia de Amazon EC2 y que se puede conectar a ella. Para obtener instrucciones al respecto, consulte la Guía de introducción a Amazon EC2.

Una instancia de Amazon EC2 se puede conectar a un nodo de clúster solo si lo ha autorizado.

Encontrar los puntos de enlace de los nodos

Una vez que el clúster se encuentra en estado disponible y ha autorizado el acceso a él, puede iniciar sesión en una instancia de Amazon EC2 y conectarse al clúster. Para ello, primero debe determinar el punto de conexión.

Búsqueda de puntos de conexión de un clúster de Valkey (modo de clúster deshabilitado) (consola)

Si un clúster de Valkey (modo de clúster deshabilitado) solo tiene un nodo, el punto de conexión de dicho nodo es el que se utiliza tanto para las operaciones de lectura como de escritura. Si un clúster tiene varios nodos, hay tres tipos de puntos de conexión: el punto de conexión primario, el punto de conexión de lectura y los puntos de conexión de nodo.

El punto de conexión principal es un nombre de DNS que siempre se resuelve en el nodo principal del clúster. El punto de conexión principal es inmune a los cambios en el clúster, como la promoción de una réplica de lectura a la función principal. Para una actividad de escritura, recomendamos que sus aplicaciones se conecten al punto de conexión principal.

Un punto de conexión del lector dividirá de manera uniforme las conexiones entrantes al punto de conexión entre todas las réplicas de lectura de un clúster de ElastiCache. Factores adicionales como cuando la aplicación crea las conexiones o cómo la aplicación (re)utiliza las conexiones determinará la distribución del tráfico. Los puntos de enlace de lector se mantienen actualizados con los cambios del clúster en tiempo real cuando se añaden o eliminan réplicas. Puede colocar las diferentes réplicas de lectura del clúster de ElastiCache en distintas zonas de disponibilidad (AZ) de AWS para garantizar la alta disponibilidad de los puntos de conexión del lector.

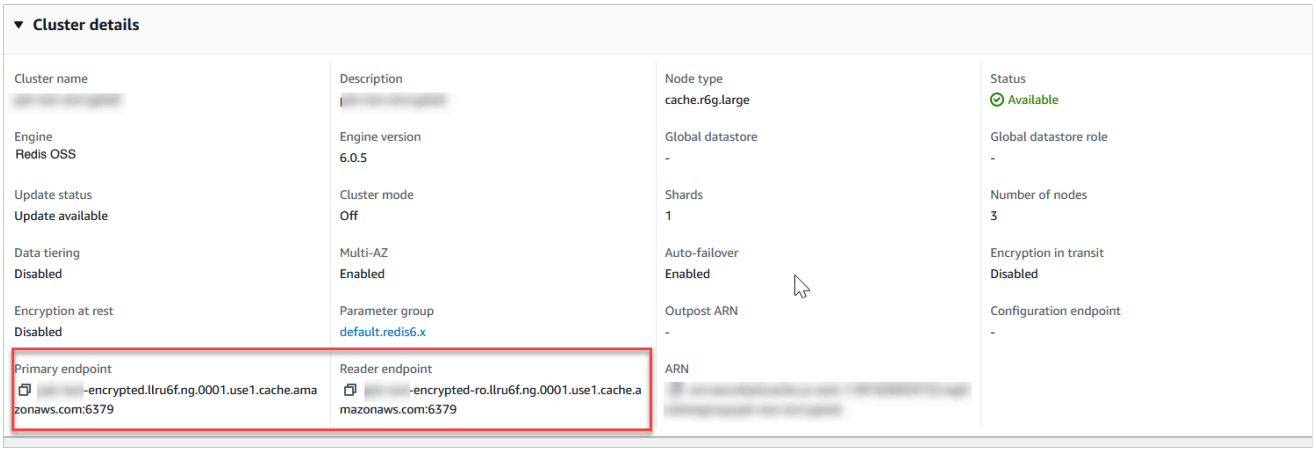
> **Note**
>
> Un punto de conexión del lector no es un equilibrador de carga. Se trata de un registro de DNS que se resolverá en una dirección IP de uno de los nodos de réplica en forma de turno rotativo.

Para la actividad de lectura, las aplicaciones pueden conectarse también a cualquier nodo del clúster. A diferencia del punto de conexión principal, los puntos de conexión de nodo se resuelven en puntos de enlace específicos. Si realiza un cambio en su clúster, como añadir o eliminar una réplica, debe actualizar los puntos de enlace del nodo en su aplicación.

Búsqueda de puntos de conexión de un clúster de Valkey (modo de clúster deshabilitado)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, seleccione Cachés de Valkey o Cachés de Redis OSS.

   Aparecerá la pantalla de clústeres con una lista que incluirá las cachés sin servidor de Valkey o Redis OSS existentes, los clústeres de Valkey (modo de clúster deshabilitado) y de Valkey (modo de clúster habilitado). Elija el clúster que ha creado en la sección Creación de un clúster de Valkey (modo de clúster deshabilitado) (consola).
3. Para buscar los puntos de conexión principal o de lector, elija el nombre del clúster (no el botón de opción).

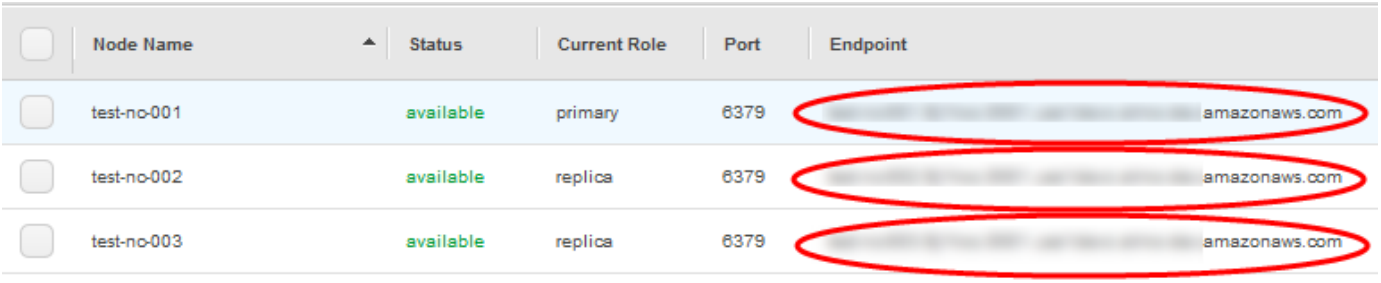


Puntos de conexión del lector y principales para un clúster de Valkey (modo de clúster deshabilitado)

Si el clúster tiene solo un nodo, quiere decir que no hay punto de conexión principal y, por lo tanto, puede continuar con el paso siguiente.

4. Si el clúster de Valkey (modo de clúster deshabilitado) tiene nodos de réplica, puede encontrar los puntos de conexión del nodo de réplica del clúster eligiendo el nombre del clúster y, a continuación, la pestaña Nodos.

   Aparecerá la pantalla de nodos con los nodos del clúster, principal y réplicas; cada uno con su punto de conexión.



   Punto de conexión del nodo de un clúster de Valkey (modo de clúster deshabilitado)

5. Para copiar un punto de conexión al portapapeles:

   a. Un punto de conexión cada vez, encuentre el punto de conexión que desea copiar.

   b. Elija el icono de copia directamente delante del punto de conexión.

   El punto de conexión se ha copiado ahora en el portapapeles. Para obtener información sobre el uso del punto de conexión a fin de conectarse a un nodo, consulte Conexión a los nodos.

Un punto de conexión principal de Valkey (modo de clúster deshabilitado) tiene un aspecto parecido al siguiente. Existe una diferencia dependiendo de si está habilitado el cifrado en tránsito o no lo está.

Cifrado en tránsito no habilitado

```
clusterName.xxxxxx.nodeId.regionAndAz.cache.amazonaws.com:port

redis-01.7abc2d.0001.usw2.cache.amazonaws.com:6379
```

Cifrado en tránsito habilitado

```
master.clusterName.xxxxxx.regionAndAz.cache.amazonaws.com:port

master.ncit.ameaqx.use1.cache.amazonaws.com:6379
```

Para buscar puntos de enlace, consulte el tema correspondiente sobre el motor y el tipo de clúster que ejecuta.

- Búsqueda de puntos de conexión en ElastiCache
- Búsqueda de puntos de conexión para un clúster de Valkey o Redis OSS (modo de clúster habilitado) (consola): necesita el punto de conexión de configuración del clúster.
- Búsqueda de puntos de conexión (AWS CLI)
- Búsqueda de puntos de enlace (API de ElastiCache)

Conexión a un grupo de replicación o a un clúster de Valkey o Redis OSS (Linux)

Ahora que tiene el punto de conexión que necesita, puede iniciar sesión en una instancia EC2 y conectarse al clúster o grupo de reproducción. En el siguiente ejemplo, se conecta a un clúster con la utilidad valkey-cli. La versión más reciente de valkey-cli también admite SSL/TLS a fin de conectar clústeres habilitados para cifrado y autenticación.

En el siguiente ejemplo se utilizan instancias de Amazon EC2 que ejecutan Amazon Linux y Amazon Linux 2. Para obtener más información sobre la instalación y compilación de valkey-cli con otras distribuciones de Linux, consulte la documentación del sistema operativo específico.

> **Note**
>
> En este proceso, se cubre la prueba de una conexión mediante la utilidad valkey-cli solo para uso no planificado. Para obtener una lista de los clientes compatibles con Valkey y Redis OSS, consulte la documentación de Valkey. Para ver ejemplos de uso de SDK de AWS con ElastiCache, consulte Tutoriales: introducción a Python y ElastiCache.

Conexión a un clúster sin cifrar en modo de clúster deshabilitado

1. Ejecute el siguiente comando para conectarse al clúster y reemplace *punto de conexión principal* y *número de puerto* por el punto de conexión del clúster y el número de puerto. (El puerto predeterminado para Valkey o Redis OSS es 6379).

   ```
   src/valkey-cli -h primary-endpoint -p port number
   ```

   El resultado en un símbolo del sistema de Valkey o Redis OSS es similar al siguiente:

   ```
   primary-endpoint:port number
   ```

2. Ahora puede ejecutar los comandos de Valkey o Redis OSS.

   ```
   set x Hello
   OK

   get x
   "Hello"
   ```

Conexión a un clúster en modo de clúster sin cifrar habilitado

1. Ejecute el siguiente comando para conectarse al clúster y reemplace *punto de conexión de configuración* y *número de puerto* por el punto de conexión del clúster y el número de puerto. (El puerto predeterminado para Valkey o Redis OSS es 6379).

   ```
   src/valkey-cli -h configuration-endpoint -c -p port number
   ```

   > **Note**
   >
   > En el comando anterior, la opción -c habilita el modo clúster en función de las redirecciones -ASK y -MOVED.

   El resultado en un símbolo del sistema de Valkey o Redis OSS es similar al siguiente:

   ```
   configuration-endpoint:port number
   ```

2. Ahora puede ejecutar los comandos de Valkey o Redis OSS. Tenga en cuenta que la redirección se produce porque la habilitó mediante la opción -c. Si la redirección no se encuentra habilitada, el comando devuelve el error MOVED (MOVIDO). Para obtener más información sobre el error MOVED, consulte la especificación del clúster de Redis OSS.

```
set x Hi
-> Redirected to slot [16287] located at 172.31.28.122:6379
OK
set y Hello
OK
get y
"Hello"
set z Bye
-> Redirected to slot [8157] located at 172.31.9.201:6379
OK
get z
"Bye"
get x
-> Redirected to slot [16287] located at 172.31.28.122:6379
"Hi"
```

Conexión a un clúster habilitado para cifrado y autenticación

De forma predeterminada, valkey-cli utiliza una conexión de TCP sin cifrar cuando se conecta a Valkey o Redis OSS. La opción `BUILD_TLS=yes` habilita SSL/TLS en el momento de la compilación de valkey-cli, como se muestra en la sección Descarga y configuración del acceso a la línea de comandos anterior. La habilitación de AUTH es opcional. Sin embargo, debe habilitar el cifrado en tránsito para habilitar AUTH. Para obtener más detalles sobre el cifrado y la autenticación de ElastiCache, consulte Cifrado en tránsito (TLS) de ElastiCache.

> **Note**
>
> Puede utilizar la opción `--tls` con valkey-cli para conectarse a clústeres cifrados en los modos de clúster habilitado y deshabilitado. Si un clúster tiene un conjunto de tokens AUTH, puede utilizar la opción `-a` para proporcionar una contraseña de AUTH.

En los siguientes ejemplos, asegúrese de reemplazar *punto de conexión de clúster* y *número de puerto* por el punto de conexión del clúster y el número de puerto. (El puerto predeterminado para Valkey o Redis OSS es 6379).

Conectarse a clústeres cifrados en modo de clúster deshabilitado

El siguiente ejemplo se conecta a un clúster habilitado para cifrado y autenticación:

```
src/valkey-cli -h cluster-endpoint --tls -a your-password -p port number
```

El siguiente ejemplo se conecta a un clúster que solo tiene habilitado el cifrado:

```
src/valkey-cli -h cluster-endpoint --tls -p port number
```

Conectarse a clústeres cifrados en modo de clúster habilitado

El siguiente ejemplo se conecta a un clúster habilitado para cifrado y autenticación:

```
src/valkey-cli -c -h cluster-endpoint --tls -a your-password -p port number
```

El siguiente ejemplo se conecta a un clúster que solo tiene habilitado el cifrado:

```
src/valkey-cli -c -h cluster-endpoint --tls -p port number
```

Después de conectarse al clúster, puede ejecutar los comandos de Valkey o Redis OSS como se muestra en los ejemplos anteriores para clústeres sin cifrar.

Alternativa a valkey-cli

Si el clúster no se encuentra en el modo de clúster habilitado y necesita realizar una conexión con el clúster para hacer una prueba corta, pero sin pasar por la compilación de valkey-cli, puede utilizar telnet u openssl. En los siguientes comandos de ejemplo, asegúrese de reemplazar *punto de conexión de clúster* y *número de puerto* por el punto de conexión del clúster y el número de puerto. (El puerto predeterminado para Valkey o Redis OSS es 6379).

El siguiente ejemplo se conecta a un clúster en modo de clúster deshabilitado que se encuentra habilitado para el cifrado o la autenticación:

```
openssl s_client -connect cluster-endpoint:port number
```

Si el clúster tiene una contraseña establecida, primero conéctese al clúster. Después de conectarse, autentique el clúster con el siguiente comando y, a continuación, presione la tecla `Enter`. En el siguiente ejemplo, reemplace *your-password* con la contraseña del clúster.

```
Auth your-password
```

El siguiente ejemplo se conecta a un clúster en modo de clúster deshabilitado que no cuenta con el cifrado o la autenticación habilitados:

```
telnet cluster-endpoint port number
```

Conexión a un grupo de replicación o a un clúster de Valkey o Redis OSS (Windows)

Para conectarse al clúster de Valkey o Redis OSS desde una instancia de EC2 de Windows utilizando la CLI de Valkey o la CLI de Redis OSS, debe descargar el paquete valkey-cli y utilizar valkey-cli.exe para conectarse al clúster de Valkey o Redis OSS desde una instancia de EC2 de Windows.

En el ejemplo siguiente, use la utilidad valkey-cli para conectarse a un clúster que no está habilitado para cifrado y ejecuta Valkey o Redis OSS. Para obtener más información sobre Valkey o Redis OSS y los comandos disponibles, consulte los comandos de Valkey y Redis OSS en el sitio web de Valkey.

Conexión a un clúster de Valkey o Redis OSS que no tiene habilitado el cifrado mediante valkey-cli

1. Conéctese a la instancia de Amazon EC2 mediante la utilidad de conexión que prefiera. Para obtener instrucciones sobre cómo conectarse a una instancia de Amazon EC2, consulte la Guía de introducción de Amazon EC2.
2. Copie y pegue el enlace https://github.com/microsoftarchive/redis/releases/download/win-3.0.504/Redis-x64-3.0.504.zip en un navegador de Internet para descargar el archivo zip para el cliente de Valkey desde una versión disponible en GitHub https://github.com/microsoftarchive/redis/releases/tag/win-3.0.504

   Extraiga el archivo zip en la carpeta/ruta deseada.

   Abra el símbolo del sistema, cambie al directorio de Valkey y ejecute el comando `c:\Valkey>valkey-cli -h Redis_Cluster_Endpoint -p 6379`.

   Por ejemplo:

```
c:\Valkey>valkey-cli -h cmd.xxxxxxx.ng.0001.usw2.cache.amazonaws.com -p 6379
```

3. Ejecute los comandos de Valkey o Redis OSS.

   Ya está conectado al clúster y puede ejecutar los comandos de Valkey o Redis OSS, como los siguientes.

```
set a "hello"          // Set key "a" with a string value and no expiration
OK
get a                  // Get value for key "a"
"hello"
get b                  // Get value for key "b" results in miss
(nil)
set b "Good-bye" EX 5  // Set key "b" with a string value and a 5 second expiration
"Good-bye"
get b                  // Get value for key "b"
"Good-bye"
                       // wait >= 5 seconds
get b
(nil)                  // key has expired, nothing returned
quit                   // Exit from valkey-cli
```

¿Qué tengo que hacer ahora?

Ahora que ha probado el ejercicio de introducción, puede explorar las siguientes secciones para obtener más información acerca de ElastiCache y las herramientas disponibles:

- Introducción a AWS
- Herramientas para Amazon Web Services
- AWS Interfaz de línea de comandos
- Referencia de la API de Amazon ElastiCache

Después de completar el ejercicio de introducción, puede leer estas secciones para obtener más información acerca de la administración de ElastiCache:

- Elección del tamaño del nodo

Desea que la caché sea lo suficientemente grande como para alojar todos los datos que necesite almacenar en caché. Al mismo tiempo, no desea pagar más caché de la que necesita. Utilice este tema para elegir el mejor tamaño de nodo.

- Procedimientos recomendados y estrategias de almacenamiento en caché de ElastiCache

  Explica cómo identificar y abordar los problemas que pueden influir en la eficiencia del clúster.

Creación de un clúster de ElastiCache basado en nodos para Redis OSS

A continuación, se indican las medidas puntuales que debe tomar para crear un clúster de ElastiCache basado en nodos para Redis OSS.

Para obtener más información sobre la configuración de ElastiCache, consulte Configuración de ElastiCache.

Temas

- Paso 1: crear un grupo de subredes
- Paso 2: crear un clúster
- Paso 3: autorizar acceso al clúster
- Paso 4: conectarse al nodo de un clúster

Paso 1: crear un grupo de subredes

Antes de crear el clúster, asegúrese de crear un grupo de subredes. Un grupo de subredes de caché es una colección de subredes que podría designar para sus clústeres en una VPC. Al lanzar un clúster en una VPC, tiene que seleccionar un grupo de subredes de caché. A continuación, ElastiCache utilizará dicho grupo de subredes de caché a fin de asignar direcciones IP de esta subred para cada nodo de caché del clúster.

Cuando cree un nuevo grupo de subredes, tenga en cuenta el número de direcciones IP disponibles. Si la subred tiene pocas direcciones IP libres, el número de nodos que podrá agregar al clúster será limitado. Para solucionar este problema, puede asignar una o varias subredes a un grupo de subredes para, de este modo, disponer de suficientes direcciones IP en la zona de disponibilidad de su clúster. Hecho esto, podrá agregar más nodos a su clúster.

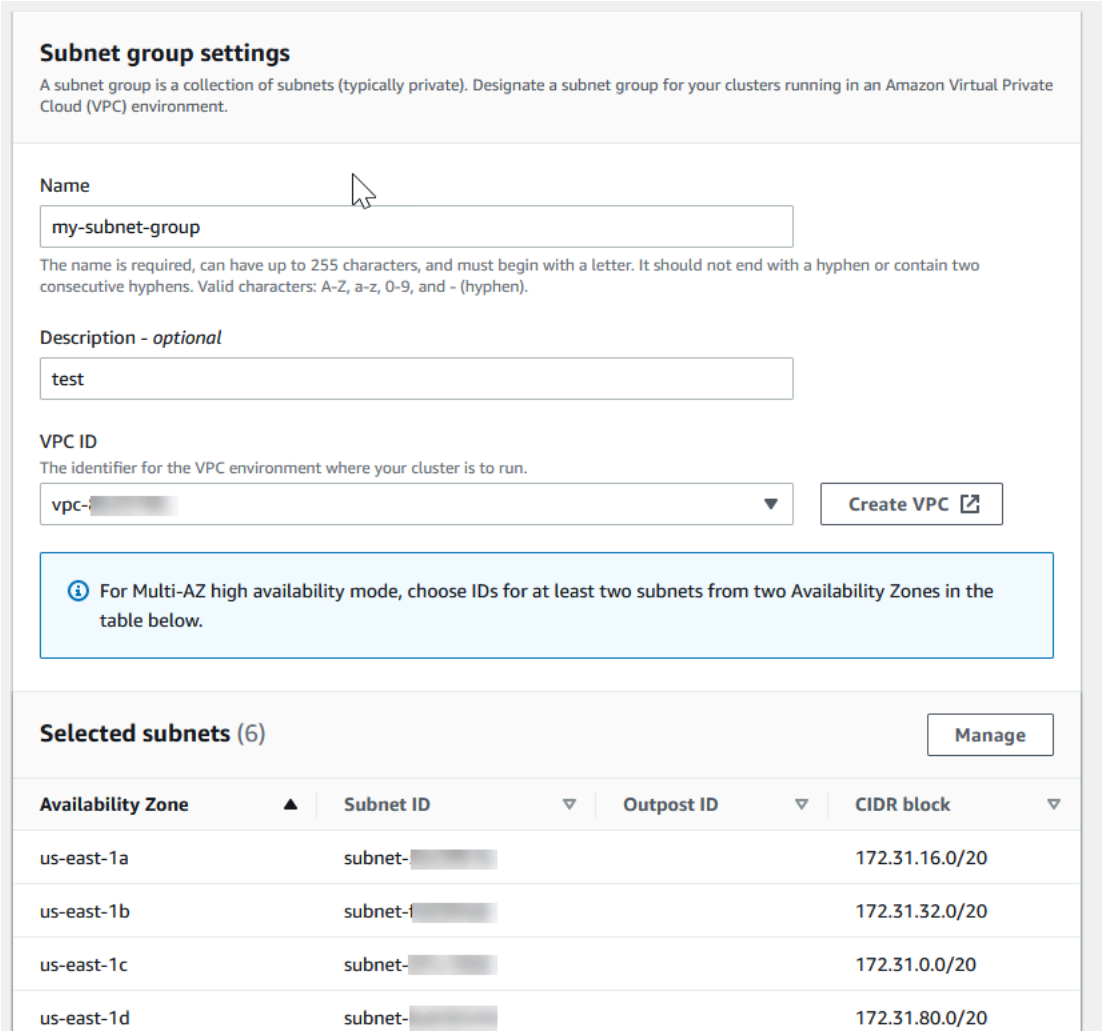
En los siguientes procedimientos, se muestra cómo crear un grupo de subredes denominado `mysubnetgroup` (consola) y la AWS CLI.

Creación de un grupo de subredes (consola)

En el siguiente procedimiento, se muestra cómo crear un grupo de subredes (consola).

Para crear un grupo de subredes (consola)

1. Inicie sesión en la AWS Management Console y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En la lista de navegación, elija Subnet Groups (Grupos de subredes).
3. Elija Create Subnet Group (Crear grupo de subredes).
4. En el asistente Create Subnet Group(Crear grupo de subredes), haga lo siguiente. Una vez que la configuración sea la deseada, elija Yes, Create (Sí, crear).
    a. En el cuadro Name (Nombre), escriba un nombre para el grupo de subredes.
    b. En el cuadro Description(Descripción), escriba la descripción del grupo de subredes.
    c. En el cuadro de VPC ID (ID de la VPC), elija la Amazon VPC que creó anteriormente.
    d. En las listas de Availability Zone (Zona de disponibilidad) y Subnet ID (ID de la subred), elija la zona de disponibilidad o zona local y el ID de su subred privada y, a continuación, elija Add (Agregar).



5. En el mensaje de confirmación que aparece, elija Close (Cerrar).

El grupo de subredes nuevo aparecerá en la lista Subnet Groups (Grupos de subredes) de la consola de ElastiCache. En la parte inferior de la ventana, podrá elegir el grupo de subredes para ver detalles tales como todas las subredes asociadas al grupo.

Crear un grupo de subredes (AWS CLI)

En el símbolo del sistema, utilice el comando `create-cache-subnet-group` para crear un grupo de subredes.

Para Linux, macOS o Unix:

```
aws elasticache create-cache-subnet-group \
    --cache-subnet-group-name mysubnetgroup \
    --cache-subnet-group-description "Testing" \
```

```
    --subnet-ids subnet-53df9c3a
```

Para Windows:

```
aws elasticache create-cache-subnet-group ^
    --cache-subnet-group-name mysubnetgroup ^
    --cache-subnet-group-description "Testing" ^
    --subnet-ids subnet-53df9c3a
```

Este comando debería producir un resultado similar al siguiente:

```
{
    "CacheSubnetGroup": {
        "VpcId": "vpc-37c3cd17",
        "CacheSubnetGroupDescription": "Testing",
        "Subnets": [
            {
                "SubnetIdentifier": "subnet-53df9c3a",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2a"
                }
            }
        ],
        "CacheSubnetGroupName": "mysubnetgroup"
    }
}
```

Para obtener más información, consulte el tema AWS CLI de la create-cache-subnet-group.

Paso 2: crear un clúster

Antes de crear un clúster para su uso en producción, obviamente debe considerar cómo configurará el clúster a fin de satisfacer las necesidades del negocio. Estos problemas se abordan en la sección Preparación de un clúster en ElastiCache. A los efectos de este ejercicio de introducción, creará un clúster con el modo de clúster deshabilitado y podrá aceptar los valores de configuración predeterminados donde se apliquen.

El clúster que crea se ejecutará en un entorno real, no en uno de pruebas. Deberá pagar las tarifas de uso estándares de ElastiCache para la instancia hasta que la elimine. Los cargos totales serán mínimos (normalmente menos de un dólar) si completa el ejercicio descrito aquí de una vez y elimina

el clúster al finalizar. Para obtener más información sobre las tarifas de uso de ElastiCache, consulte Amazon ElastiCache.

El clúster se lanza en una nube virtual privada (VPC) en función del servicio de Amazon VPC.

Creación de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) (consola)

Creación de un clúster de Redis OSS (modo de clúster deshabilitado) con la consola de ElastiCache

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En la lista de la esquina superior derecha, elija la región de AWS en la que desea lanzar este clúster.
3. En el panel de navegación, seleccione Get started (Introducción).
4. Elija Create VPC (Crear una VPC) y siga los pasos que se detallan en Creating a Virtual Private Cloud (VPC) (Creación de una nube privada virtual).
5. En la página del panel de ElastiCache, elija Caché de Valkey o Caché de Redis OSS. Para este ejercicio, seleccionaremos Caché de Redis OSS y, a continuación, seleccionaremos Crear caché de Redis OSS.
6. En Cluster settings (Configuración de clúster), realice la siguiente operación:

    a. Elija Configure and create a new cluster (Configurar y crear un nuevo clúster).

    b. Para Cluster mode (Modo clúster), elige Disabled (Desactivado).

    c. Para Cluster info (Información del clúster) introduzca un valor para Name (Nombre).

    d. (Opcional) Introduzca un valor para Description (Descripción).
7. En Location (Ubicación):

    AWS Cloud

    1. Para Nube de AWS, le recomendamos que acepte la configuración predeterminada de Multi-AZ y Conmutación por error automática. Para obtener más información, consulte Minimización del tiempo de inactividad en ElastiCache para Redis OSS con Multi-AZ.
    2. En Cluster settings (Configuración del clúster)

        a. Para Engine version (Versión del motor), elija una versión disponible.

        b. En Port (Puerto), utilice el puerto predeterminado, 6379. Si tiene un motivo para utilizar un puerto diferente, escriba el número de puerto.

c. En Grupo de parámetros, elija un grupo de parámetros o cree uno nuevo. Los grupos de parámetros controlan los parámetros de tiempo de ejecución de su clúster. Para obtener más información acerca de los grupos de parámetros, consulte Parámetros de Valkey y Redis OSS y Creación de un grupo de parámetros de ElastiCache.

> **Note**
>
> Cuando se selecciona un grupo de parámetros para definir los valores de configuración del motor, dicho grupo de parámetros se aplica a todos los clústeres del almacén de datos global. En la página Parameter Groups (Grupos de parámetros), el atributo Global sí/no, indica si un grupo de parámetros forma parte de un almacén de datos global.

d. Para Node type, elija la flecha hacia abajo
(▼).
En el cuadro de diálogo Cambiar tipo de nodo, elija un valor para Familia de instancias para el tipo de nodo que desea. A continuación, elija el tipo de nodo que desea utilizar para este clúster y, a continuación, elija Guardar.

Para obtener más información, consulte Elección del tamaño del nodo.

Si elige un tipo de nodo r6gd, la organización de datos en niveles se activará automáticamente. Para obtener más información, consulte Organización de datos en niveles en ElastiCache.

e. Para Number of replicas (Número de réplicas), elija el número de réplicas de lectura que desee. Si habilitó Multi-AZ, el número debe estar entre 1 y 5.

3. En Connectivity (Conectividad)

a. Para el Network type (Tipo de red), elija las versiones IP que admitirá este clúster.

b. En Subnet groups (Grupos de subredes), elija la subred que desea aplicar a este clúster. ElastiCache utiliza dicho grupo de subredes para elegir una subred y direcciones IP pertenecientes a dicha subred a fin de asociarlas a sus nodos. Los clústeres de ElastiCache requieren una subred de doble pila con direcciones IPv4 e IPv6 asignadas para funcionar en modo de doble pila y una subred solo IPv6 para funcionar como solo IPv6.

Al crear un nuevo grupo de subredes, escriba el VPC ID (ID de VPC) al que pertenece.

Para obtener más información, consulte:

- Elección de un tipo de red en ElastiCache.
- Cree una subred en la VPC.

Si es Uso de Local Zones con ElastiCache, debe crear o elegir una subred que se encuentre en la zona local.

Para obtener más información, consulte Subredes y grupos de subredes.

4. En Availability zone placements Ubicaciones de las zonas de disponibilidad), tiene dos opciones:

   - No preference (Sin preferencia): ElastiCache elige la zona de disponibilidad.
   - Specify availability zones (Especificar zonas de disponibilidad): especifique la zona de disponibilidad para cada clúster.

     Si decide especificar las zonas de disponibilidad, para cada clúster en cada fragmento, elija la zona de disponibilidad de la lista.

   Para obtener más información, consulte Elección de regiones y zonas de disponibilidad para ElastiCache.

5. Elija Next (Siguiente).
6. En Configuración avanzada de Redis OSS

   - Para Security (Seguridad):

     i. Para cifrar sus datos, tiene las siguientes opciones:

        - Encryption at rest (Cifrado en reposo): permite el cifrado de los datos almacenados en el disco. Para obtener más información, consulte Cifrado en reposo.

> **Note**
>
> Tiene la opción de suministrar una clave de cifrado distinta al elegir Customer Managed AWS KMS key (Clave KMS de administrada por el cliente) y elegir la clave. Para obtener más información, consulte Uso de claves de AWS KMS administradas por el cliente.

- Encryption in-transit (Cifrado en tránsito): permite el cifrado de datos del cable. Para obtener más información, consulte Cifrado en tránsito. Para la versión 6.0 del motor de ElastiCache para Redis OSS y posteriores, si habilita el cifrado en tránsito, se le pedirá que especifique una de las siguientes opciones de Control de acceso:
  - No Access Control (Sin control de acceso): este es el valor predeterminado. Esto indica que no hay restricciones en el acceso de los usuarios al clúster.
  - User Group Access Control List (Lista de control de acceso del grupo de usuarios): seleccione un grupo de usuarios con un conjunto definido de usuarios que pueden acceder al clúster. Para obtener más información, consulte Administración de grupos de usuarios con la consola y la CLI.
  - Usuario predeterminado con el comando AUTH: un mecanismo de autenticación para el servidor de Valkey y Redis OSS. Para obtener más información, consulte AUTH.
- AUTH: un mecanismo de autenticación para el servidor de Redis OSS. Para obtener más información, consulte AUTH.

> **Note**
>
> Para Redis OSS 3.2.6 en adelante, excepto la versión 3.2.10, AUTH de Redis OSS es la única opción.

ii. En Grupos de seguridad, elija los grupos de seguridad que desea para este clúster. Un grupo de seguridad actúa como un firewall para controlar el acceso de red al clúster. Puede utilizar el grupo de seguridad predeterminado para la VPC o crear uno nuevo.

Para obtener más información sobre los grupos de seguridad, consulte Grupos de seguridad de su VPC en la Guía del usuario de Amazon VPC.

7. En las copias de seguridad automáticas programadas regularmente, seleccione Enable automatic backups (Habilitar copias de seguridad automáticas) y, a continuación, escriba el número de días que desea conservar cada copia de seguridad automática antes de que se elimine automáticamente. Si no quiere tener copias de seguridad automáticas programadas regularmente, desactive la casilla de verificación Enable automatic backups (Habilitar copias de seguridad automáticas). En cualquier caso, siempre tendrá la opción de crear copias de seguridad manuales.

   Para obtener más información acerca de la copia de seguridad y restauración, consulte Instantánea y restauración.

8. (Opcional) Especifique un periodo de mantenimiento. El periodo de mantenimiento hace alusión al tiempo, generalmente una hora, de cada semana en la que ElastiCache programa el mantenimiento del sistema en su clúster. Puede permitir que ElastiCache elija el día y la hora de su periodo de mantenimiento (No preference [Sin preferencia]), o bien puede elegir el día, la hora y la duración por su cuenta (Specify maintenance window [Especificar periodo de mantenimiento]). Si elige Especificar el periodo de mantenimiento en las listas, seleccione Día de inicio, Hora de inicio y Duración (en horas) para el periodo de mantenimiento. Todas las horas se indican en UCT.

   Para obtener más información, consulte Administración del mantenimiento de clústeres de ElastiCache.

9. (Opcional) Para Logs (Registros):

   - En Log format (Formato de registro), elija Text o JSON.
   - En Destination Type (Tipo de destino), elija CloudWatch Logs o Kinesis Firehose.
   - En Destino de registro, elija Crear nuevo e introduzca el nombre del grupo de registro de Registros de CloudWatch o el nombre del flujo de Firehose, o elija Seleccionar existente y, a continuación, elija el nombre del grupo de registro de Registros de CloudWatch o el nombre del flujo de Firehose.

10. Para Tags (Etiquetas), para ayudarlo a administrar sus clústeres y otros recursos de ElastiCache, puede asignar sus propios metadatos a cada recurso en forma de etiquetas. Para obtener más información, consulte Etiquetado de los recursos de ElastiCache.

11. Elija Siguiente.
12. Revise todas las entradas y opciones y, a continuación, realice todos los cambios necesarios. Cuando haya terminado, elija Create (Crear).

On premises

1. Para On premises (En las instalaciones), le recomendamos que deje Auto-failover (Conmutación por error automática) habilitada. Para obtener más información, consulte Minimización del tiempo de inactividad en ElastiCache para Redis OSS con Multi-AZ.
2. Para terminar de crear el clúster, siga los pasos que se indican en Using Outposts (Uso de Outposts).

En cuanto el estado de su clúster sea available (disponible), podrá conceder a Amazon EC2 acceso a este, así como conectarse a él y comenzar a utilizarlo. Para obtener más información, consulte Paso 3. Autorización del acceso al clúster y Paso 4. Conexión al nodo de un clúster.

> **Important**
>
> Una vez que su clúster esté disponible, se cobrará por cada hora u hora parcial que el clúster esté activo, incluso si no lo está utilizando de forma activa. Para dejar de incurrir en cargos por este clúster, debe eliminarlo. Consulte Eliminación de un clúster en ElastiCache.

Creación de un clúster de Redis OSS (modo de clúster deshabilitado) (AWS CLI)

Example

El siguiente código de la CLI crea un clúster de Redis OSS (modo de clúster deshabilitado) sin réplicas.

Para Linux, macOS o Unix:

```
aws elasticache create-cache-cluster \
--cache-cluster-id my-cluster \
--cache-node-type cache.r4.large \
--engine redis \
--num-cache-nodes 1 \
--snapshot-arns arn:aws:s3:::my_bucket/snapshot.rdb
```

Para Windows:

```
aws elasticache create-cache-cluster ^
--cache-cluster-id my-cluster ^
--cache-node-type cache.r4.large ^
--engine redis ^
--num-cache-nodes 1 ^
--snapshot-arns arn:aws:s3:::my_bucket/snapshot.rdb
```

Para trabajar con el modo de clúster habilitado, consulte los temas siguientes:

- Para utilizar la consola, consulte Creación de un clúster de Valkey o Redis OSS (modo de clúster habilitado) (consola).
- Para utilizar la AWS CLI, consulte Creación de un clúster de Valkey o Redis OSS (modo de clúster habilitado) (AWS CLI).

Paso 3: autorizar acceso al clúster

En esta sección se presupone que se encuentra familiarizado con los procesos de lanzar instancias de Amazon EC2 y conectarse a estas. Para obtener más información, consulte la Guía de introducción de Amazon EC2.

Todos los clústeres de ElastiCache se han diseñado para que se pueda acceder a estos desde una instancia de Amazon EC2. Lo más común es obtener acceso a un clúster de ElastiCache desde una instancia de Amazon EC2 en la misma Amazon Virtual Private Cloud (Amazon VPC), este será el caso utilizado para este ejercicio.

De forma predeterminada, el acceso de red al clúster está limitado a la cuenta que se utilizó para crearlo. Antes de poder conectarse a un clúster desde una instancia EC2, debe autorizar a la instancia EC2 el acceso al clúster. Los pasos necesarios dependerán de si ha lanzado el clúster en EC2-VPC o EC2-Classic.

El caso de uso más común es cuando una aplicación implementada en una instancia EC2 debe conectarse a un clúster en la misma VPC. La forma más sencilla de administrar el acceso entre instancias EC2 y clústeres de la misma VPC es realizar lo siguiente:

1. Cree un grupo de seguridad de VPC para su clúster. Este grupo de seguridad se puede utilizar para restringir el acceso a las instancias de clúster. Por ejemplo, puede crear una regla personalizada para este grupo de seguridad que permita el acceso mediante TCP utilizando el puerto que asignó al clúster de base de datos cuando lo creó y una dirección IP que se utilizará para obtener acceso al clúster.

   El puerto predeterminado para los clústeres y grupos de replicación de Redis OSS es 6379.

   > **Important**
   >
   > Los grupos de seguridad de Amazon ElastiCache solo son aplicables a los clústeres que no se ejecutan en un entorno de Amazon Virtual Private Cloud (VPC). Si la ejecución tiene lugar en una Amazon Virtual Private Cloud, Security Groups (Grupos de seguridad) no se encontrará disponible en el panel de navegación de la consola.
   > Si ejecuta los nodos de ElastiCache en una Amazon VPC, podrá controlar el acceso a los clústeres con los grupos de seguridad de Amazon VPC, que son distintos de los grupos de seguridad de ElastiCache. Para obtener más información sobre la utilización de ElastiCache en Amazon VPC, consulte Seguridad de ElastiCache y Amazon VPC

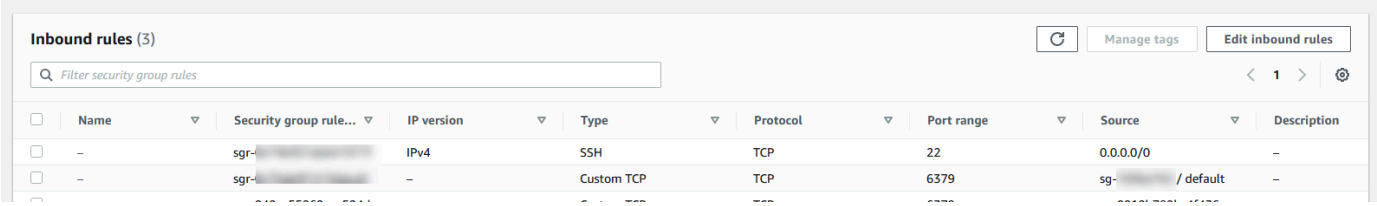
2. Cree un grupo de seguridad de VPC para sus instancias EC2 (servidores web y de aplicaciones). Si es necesario, este grupo de seguridad puede permitir el acceso a la instancia EC2 desde Internet a través de la tabla de enrutamiento de la VPC. Por ejemplo, puede establecer reglas en este grupo de seguridad para permitir el acceso mediante TCP a la instancia EC2 a través del puerto 22.
3. Cree reglas personalizadas en el grupo de seguridad para su clúster que permitan las conexiones desde el grupo de seguridad que creó para las instancias EC2. Esto permitirá a cualquier miembro del grupo de seguridad obtener acceso a los clústeres.

> **Note**
>
> Si tiene previsto utilizar Uso de Local Zones con ElastiCache, asegúrese de que las ha habilitado. Cuando crea un grupo de subredes en esa zona local, la VPC se amplía a dicha zona local y la VPC tratará la subred como cualquier subred de cualquier otra zona de disponibilidad. Todas las gateways y tablas de enrutamiento relevantes se ajustarán de forma automática.

Para crear una regla en un grupo de seguridad de VPC que permita establecer conexiones desde otro grupo de seguridad

1. Inicie sesión en la consola de administración de AWS y abra la consola de Amazon VPC en https://console.aws.amazon.com/vpc.
2. En el panel de navegación, elija Security Groups (Grupos de seguridad).
3. Seleccione o cree un grupo de seguridad que utilizará para sus instancias de clúster. En Inbound Rules (Reglas de entrada), seleccione Edit Inbound Rules (Editar reglas de entrada) y, a continuación, seleccione Add Rule (Agregar regla). Este grupo de seguridad permitirá el acceso a los miembros de otro grupo de seguridad.
4. En Type (Tipo), elija Custom TCP Rule (Personalizar regla de TCP).
   a. En Port Range (Rango de puerto), especifique el puerto que utilizó al crear su clúster.

      El puerto predeterminado para los clústeres y grupos de replicación de Redis OSS es 6379.
   b. En el cuadro Source (Fuente), comience a escribir el ID del grupo de seguridad. Desde la lista, seleccione el grupo de seguridad que utilizará para sus instancias de Amazon EC2.
5. Cuando haya terminado, elija Save (Guardar).



Una vez que haya habilitado el acceso, se encontrará listo para conectarse al nodo, como se describe en la siguiente sección.

Para obtener más información sobre cómo obtener acceso al clúster de ElastiCache desde una Amazon VPC diferente, desde una región de AWS diferente o incluso desde la red de la empresa, consulte lo siguiente:

- Patrones de acceso para obtener acceso a una caché de ElastiCache en una Amazon VPC
- Acceso a recursos de ElastiCache desde fuera de AWS

Paso 4: conectarse al nodo de un clúster

Antes de continuar, realice el Paso 3: autorizar acceso al clúster.

En esta sección se da por sentado que ha creado una instancia de Amazon EC2 y que se puede conectar a ella. Para obtener instrucciones al respecto, consulte la Guía de introducción a Amazon EC2.

Una instancia de Amazon EC2 se puede conectar a un nodo de clúster solo si lo ha autorizado.

Encontrar los puntos de enlace de los nodos

Una vez que el clúster se encuentra en estado disponible y ha autorizado el acceso a él, puede iniciar sesión en una instancia de Amazon EC2 y conectarse al clúster. Para ello, primero debe determinar el punto de conexión.

Búsqueda de puntos de conexión de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) (consola)

Si un clúster de Redis OSS (modo de clúster deshabilitado) solo tiene un nodo, el punto de conexión de dicho nodo es el que se utiliza tanto para las operaciones de lectura como de escritura. Si un clúster tiene varios nodos, hay tres tipos de puntos de conexión: el punto de conexión primario, el punto de conexión de lectura y los puntos de conexión de nodo.

El punto de conexión principal es un nombre de DNS que siempre se resuelve en el nodo principal del clúster. El punto de conexión principal es inmune a los cambios en el clúster, como la promoción de una réplica de lectura a la función principal. Para una actividad de escritura, recomendamos que sus aplicaciones se conecten al punto de conexión principal.

Un punto de conexión del lector dividirá de manera uniforme las conexiones entrantes al punto de conexión entre todas las réplicas de lectura de un clúster de ElastiCache para Redis OSS. Factores adicionales como cuando la aplicación crea las conexiones o cómo la aplicación (re)utiliza las conexiones determinará la distribución del tráfico. Los puntos de enlace de lector se mantienen actualizados con los cambios del clúster en tiempo real cuando se añaden o eliminan réplicas. Puede colocar las diferentes réplicas de lectura del clúster de ElastiCache para Redis OSS en distintas zonas de disponibilidad (AZ) de AWS para garantizar la alta disponibilidad de los puntos de conexión del lector.

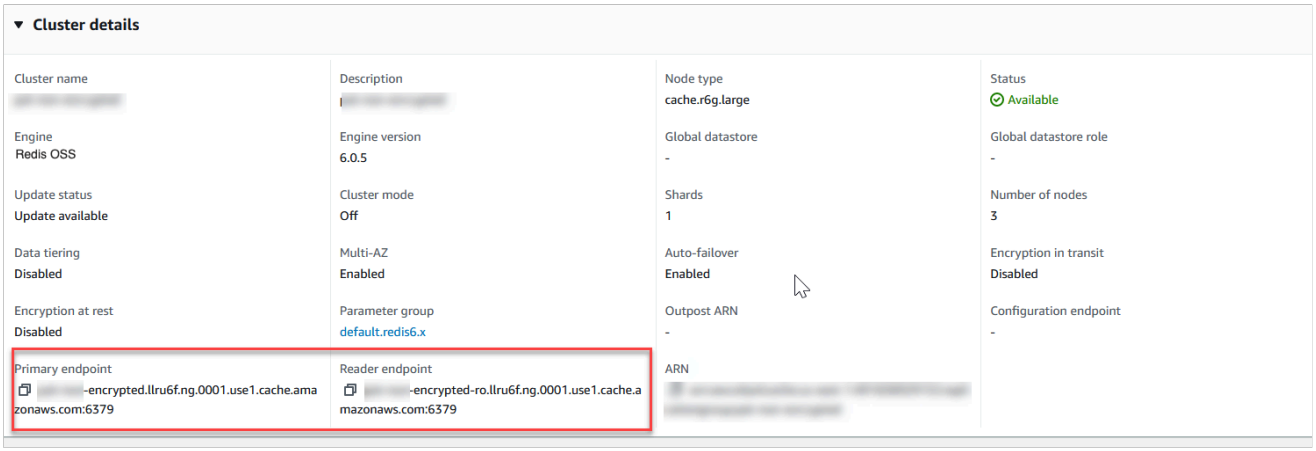
> **Note**
>
> Un punto de conexión del lector no es un equilibrador de carga. Se trata de un registro de DNS que se resolverá en una dirección IP de uno de los nodos de réplica en forma de turno rotativo.

Para la actividad de lectura, las aplicaciones pueden conectarse también a cualquier nodo del clúster. A diferencia del punto de conexión principal, los puntos de conexión de nodo se resuelven en puntos de enlace específicos. Si realiza un cambio en su clúster, como añadir o eliminar una réplica, debe actualizar los puntos de enlace del nodo en su aplicación.

Búsqueda de puntos de conexión de un clúster de Redis OSS (modo de clúster deshabilitado)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, seleccione Cachés de Redis OSS.

   Aparecerá la pantalla de clústeres con una lista que incluirá las cachés sin servidor de Valkey o Redis OSS existentes, los clústeres de Redis OSS (modo de clúster deshabilitado) y los clústeres de Redis OSS (modo de clúster habilitado). Elija el clúster que ha creado en la sección Creación de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) (consola).
3. Para buscar los puntos de conexión principal o de lector, elija el nombre del clúster (no el botón de opción).

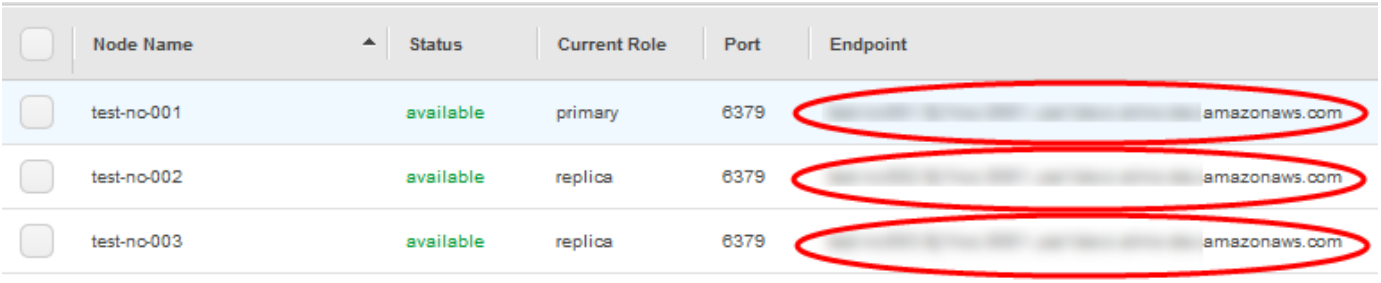


Puntos de conexión del lector y principales para un clúster de Redis OSS (modo de clúster deshabilitado)

Si el clúster tiene solo un nodo, quiere decir que no hay punto de conexión principal y, por lo tanto, puede continuar con el paso siguiente.

4. Si el clúster de Redis OSS (modo de clúster deshabilitado) tiene nodos de réplica, puede encontrar los puntos de conexión del nodo de réplica del clúster eligiendo el nombre del clúster y, a continuación, la pestaña Nodos.

   Aparecerá la pantalla de nodos con los nodos del clúster, principal y réplicas; cada uno con su punto de conexión.



   Puntos de conexión de nodo de un clúster de Redis OSS (modo de clúster deshabilitado)

5. Para copiar un punto de conexión al portapapeles:

   a. Un punto de conexión cada vez, encuentre el punto de conexión que desea copiar.

   b. Elija el icono de copia directamente delante del punto de conexión.

   El punto de conexión se ha copiado ahora en el portapapeles. Para obtener información sobre el uso del punto de conexión a fin de conectarse a un nodo, consulte Conexión a los nodos.

Un punto de conexión principal de Redis OSS (modo de clúster deshabilitado) tiene un aspecto parecido al siguiente. Existe una diferencia dependiendo de si está habilitado el cifrado en tránsito o no lo está.

Cifrado en tránsito no habilitado

```
clusterName.xxxxxx.nodeId.regionAndAz.cache.amazonaws.com:port

redis-01.7abc2d.0001.usw2.cache.amazonaws.com:6379
```

Cifrado en tránsito habilitado

```
master.clusterName.xxxxxx.regionAndAz.cache.amazonaws.com:port

master.ncit.ameaqx.use1.cache.amazonaws.com:6379
```

Para buscar puntos de enlace, consulte el tema correspondiente sobre el motor y el tipo de clúster que ejecuta.

- Búsqueda de puntos de conexión en ElastiCache
- Búsqueda de puntos de conexión para un clúster de Valkey o Redis OSS (modo de clúster habilitado) (consola): necesita el punto de conexión de configuración del clúster.
- Búsqueda de puntos de conexión (AWS CLI)
- Búsqueda de puntos de enlace (API de ElastiCache)

Conexión a un grupo de replicación o a un clúster de Valkey o Redis OSS (Linux)

Ahora que tiene el punto de conexión que necesita, puede iniciar sesión en una instancia EC2 y conectarse al clúster o grupo de reproducción. En el siguiente ejemplo, se conecta a un clúster con la utilidad valkey-cli. La versión más reciente de valkey-cli también admite SSL/TLS a fin de conectar clústeres habilitados para cifrado y autenticación.

En el siguiente ejemplo se utilizan instancias de Amazon EC2 que ejecutan Amazon Linux y Amazon Linux 2. Para obtener más información sobre la instalación y compilación de valkey-cli con otras distribuciones de Linux, consulte la documentación del sistema operativo específico.

> **Note**
>
> En este proceso, se cubre la prueba de una conexión mediante la utilidad valkey-cli solo para uso no planificado. Para obtener una lista de los clientes compatibles, consulte la documentación de Valkey. Para ver ejemplos de uso de SDK de AWS con ElastiCache, consulte Tutoriales: introducción a Python y ElastiCache.

Conexión a un clúster sin cifrar en modo de clúster deshabilitado

1. Ejecute el siguiente comando para conectarse al clúster y reemplace *punto de conexión principal* y *número de puerto* por el punto de conexión del clúster y el número de puerto. (El puerto predeterminado para Valkey y Redis OSS es 6379).

   ```
   src/valkey-cli -h primary-endpoint -p port number
   ```

   De este modo, aparecerá un símbolo del sistema similar al siguiente:

   ```
   primary-endpoint:port number
   ```

2. Ahora puede ejecutar los comandos de Valkey y Redis OSS.

   ```
   set x Hello
   OK

   get x
   "Hello"
   ```

Conexión a un clúster en modo de clúster sin cifrar habilitado

1. Ejecute el siguiente comando para conectarse al clúster y reemplace *punto de conexión de configuración* y *número de puerto* por el punto de conexión del clúster y el número de puerto. (El puerto predeterminado para Valkey y Redis OSS es 6379).

   ```
   src/valkey-cli -h configuration-endpoint -c -p port number
   ```

   > **Note**
   >
   > En el comando anterior, la opción -c habilita el modo clúster en función de las redirecciones -ASK y -MOVED.

   De este modo, aparecerá un símbolo del sistema similar al siguiente:

   ```
   configuration-endpoint:port number
   ```

2. Ahora puede ejecutar los comandos de Valkey y Redis OSS. Tenga en cuenta que la redirección se produce porque la habilitó mediante la opción -c. Si la redirección no se encuentra habilitada, el comando devuelve el error MOVED (MOVIDO). Para obtener más información sobre el error MOVED, consulte la especificación del clúster.

```
set x Hi
-> Redirected to slot [16287] located at 172.31.28.122:6379
OK
set y Hello
OK
get y
"Hello"
set z Bye
-> Redirected to slot [8157] located at 172.31.9.201:6379
OK
get z
"Bye"
get x
-> Redirected to slot [16287] located at 172.31.28.122:6379
"Hi"
```

Conexión a un clúster habilitado para cifrado y autenticación

De forma predeterminada, valkey-cli utiliza una conexión de TCP sin cifrar cuando se conecta a Valkey y Redis OSS. La opción `BUILD_TLS=yes` habilita SSL/TLS en el momento de la compilación de valkey-cli, como se muestra en la sección Descarga y configuración del acceso a la línea de comandos anterior. La habilitación de AUTH es opcional. Sin embargo, debe habilitar el cifrado en tránsito para habilitar AUTH. Para obtener más detalles sobre el cifrado y la autenticación de ElastiCache, consulte Cifrado en tránsito (TLS) de ElastiCache.

> **Note**
>
> Puede utilizar la opción `--tls` con valkey-cli para conectarse a clústeres cifrados en los modos de clúster habilitado y deshabilitado. Si un clúster tiene un conjunto de tokens AUTH, puede utilizar la opción `-a` para proporcionar una contraseña de AUTH.

En los siguientes ejemplos, asegúrese de reemplazar *punto de conexión de clúster* y *número de puerto* por el punto de conexión del clúster y el número de puerto. (El puerto predeterminado para Redis OSS es 6379).

Conectarse a clústeres cifrados en modo de clúster deshabilitado

El siguiente ejemplo se conecta a un clúster habilitado para cifrado y autenticación:

```
src/valkey-cli -h cluster-endpoint --tls -a your-password -p port number
```

El siguiente ejemplo se conecta a un clúster que solo tiene habilitado el cifrado:

```
src/valkey-cli -h cluster-endpoint --tls -p port number
```

Conectarse a clústeres cifrados en modo de clúster habilitado

El siguiente ejemplo se conecta a un clúster habilitado para cifrado y autenticación:

```
src/valkey-cli -c -h cluster-endpoint --tls -a your-password -p port number
```

El siguiente ejemplo se conecta a un clúster que solo tiene habilitado el cifrado:

```
src/valkey-cli -c -h cluster-endpoint --tls -p port number
```

Después de conectarse al clúster, puede ejecutar los comandos de Valkey o Redis OSS como se muestra en los ejemplos anteriores para clústeres sin cifrar.

Una alternativa a valkey-cli o redis-cli

Si el clúster no se encuentra en el modo de clúster habilitado y necesita realizar una conexión con el clúster para hacer una prueba corta, pero sin pasar por la compilación de valkey-cli o redis-cli, puede utilizar telnet u openssl. En los siguientes comandos de ejemplo, asegúrese de reemplazar *punto de conexión de clúster* y *número de puerto* por el punto de conexión del clúster y el número de puerto. (El puerto predeterminado para Redis OSS es 6379).

El siguiente ejemplo se conecta a un clúster en modo de clúster deshabilitado que se encuentra habilitado para el cifrado o la autenticación:

```
openssl s_client -connect cluster-endpoint:port number
```

Si el clúster tiene una contraseña establecida, primero conéctese al clúster. Después de conectarse, autentique el clúster con el siguiente comando y, a continuación, presione la tecla `Enter`. En el siguiente ejemplo, reemplace *your-password* con la contraseña del clúster.

```
Auth your-password
```

El siguiente ejemplo se conecta a un clúster en modo de clúster deshabilitado que no cuenta con el cifrado o la autenticación habilitados:

```
telnet cluster-endpoint port number
```

Conexión a un grupo de replicación o a un clúster de Valkey o Redis OSS (Windows)

Para conectarse al clúster desde una instancia de EC2 de Windows utilizando la CLI de Valkey o Redis OSS, debe descargar el paquete valkey-cli y utilizar valkey-cli.exe para conectarse al clúster de Valkey o Redis OSS desde una instancia de EC2 de Windows.

En el ejemplo siguiente, use la utilidad valkey-cli para conectarse a un clúster que no está habilitado para cifrado y ejecuta Valkey o Redis OSS. Para obtener más información sobre Valkey y los comandos disponibles, consulte los comandos de Valkey en el sitio web de Valkey.

Conexión a un clúster de Valkey o Redis OSS que no tiene habilitado el cifrado mediante valkey-cli

1. Conéctese a la instancia de Amazon EC2 mediante la utilidad de conexión que prefiera. Para obtener instrucciones sobre cómo conectarse a una instancia de Amazon EC2, consulte la Guía de introducción de Amazon EC2.
2. Copie y pegue el enlace https://github.com/microsoftarchive/redis/releases/download/win-3.0.504/Redis-x64-3.0.504.zip en un navegador de Internet para descargar el archivo zip para el cliente de Redis OSS desde una versión disponible en GitHub https://github.com/microsoftarchive/redis/releases/tag/win-3.0.504

   Extraiga el archivo zip en la carpeta/ruta deseada.

   Abra el símbolo del sistema, cambie al directorio de Valkey y ejecute el comando `c:\Valkey>valkey-cli -h Valkey_Cluster_Endpoint -p 6379`.

   Por ejemplo:

   ```
   c:\Valkey>valkey-cli -h cmd.xxxxxxx.ng.0001.usw2.cache.amazonaws.com -p 6379
   ```

3. Ejecute los comandos de Valkey o Redis OSS.

   Ya está conectado al clúster y puede ejecutar los comandos de Valkey o Redis OSS, como los siguientes.

```
set a "hello"          // Set key "a" with a string value and no expiration
OK
get a                  // Get value for key "a"
"hello"
get b                  // Get value for key "b" results in miss
(nil)
set b "Good-bye" EX 5  // Set key "b" with a string value and a 5 second expiration
"Good-bye"
get b                  // Get value for key "b"
"Good-bye"
                       // wait >= 5 seconds
get b
(nil)                  // key has expired, nothing returned
quit                   // Exit from valkey-cli
```

Eliminación de un clúster

Siempre que un clúster tenga el estado available, se cobrará por él, independientemente de si lo esté usando de forma activa o no. Para que dejen de devengarse cargos, elimine el clúster.

> **Warning**
>
> - Cuando se elimina un clúster de ElastiCache, se retienen las instantáneas manuales. También puede crear una instantánea final antes de eliminar el clúster. Por el contrario, las instantáneas de caché automáticas no se conservan. Para obtener más información, consulte Instantánea y restauración.
> - El permiso `CreateSnapshot` es necesario para crear una instantánea final. Sin este permiso, la llamada a la API fallará con una excepción `Access Denied`.

Uso de Consola de administración de AWS

El siguiente procedimiento elimina un único clúster de su implementación. Para eliminar varios clústeres, repita el procedimiento por cada clúster que desee eliminar. No es necesario esperar a un clúster para terminar de eliminarlo antes de empezar el procedimiento para eliminar otro clúster.

Para eliminar un clúster

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel del motor de ElastiCache, elija Valkey o Redis OSS.

   Se mostrará una lista de todas las cachés que se ejecutan en ese motor.
3. Para elegir el clúster que se va a eliminar, elija el nombre del clúster de la lista de clústeres. En este caso, el nombre del clúster que creó en Paso 2: crear un clúster.

   > **Important**
   >
   > Los clústeres solo se pueden eliminar uno por uno en la consola de ElastiCache. Al elegir varios clústeres, se deshabilita la operación de eliminación.
4. En Actions (Acciones), seleccione Delete (Eliminar).
5. En la pantalla de confirmación Eliminar clúster, escriba el nombre del clúster y elija Copia de seguridad final. A continuación, seleccione Eliminar para eliminar el clúster o Cancelar para conservarlo.

   Si elige Delete, el estado del clúster cambia a deleting.

En cuanto el clúster desaparezca de la lista de clústeres, dejará de incurrir en gastos.

Uso de AWS CLI

El código siguiente elimina el clúster `my-cluster`. En este caso, sustituya `my-cluster` con el nombre del clúster que creó en Paso 2: crear un clúster.

```
aws elasticache delete-cache-cluster --cache-cluster-id my-cluster
```

La acción `delete-cache-cluster` de la CLI solo elimina un clúster. Para eliminar varios clústeres, llame a `delete-cache-cluster` por cada clúster de caché que desee eliminar. No es necesario esperar a que se termine de eliminar un clúster antes de eliminar otro.

Para Linux, macOS o Unix:

```
aws elasticache delete-cache-cluster \
    --cache-cluster-id my-cluster \
    --region us-east-2
```

Para Windows:

```
aws elasticache delete-cache-cluster ^
    --cache-cluster-id my-cluster ^
    --region us-east-2
```

Para obtener más información, consulte la AWS CLI para el tema de ElastiCache `delete-cache-cluster`.

Otros tutoriales y vídeos de ElastiCache

Los siguientes tutoriales abordan tareas de interés para el usuario de Amazon ElastiCache.

- Videos de ElastiCache
- Tutorial: Configuring a Lambda Function to Access Amazon ElastiCache in an Amazon VPC

Videos de ElastiCache

A continuación, puede encontrar vídeos que lo ayudarán a conocer los conceptos básicos y avanzados de Amazon ElastiCache. Para obtener información sobre la formación de AWS, consulte AWS Training & Certification.

Temas

- Videos introductorios
- Videos avanzados

Videos introductorios

Los siguientes videos presentan Amazon ElastiCache.

Temas

- AWS re:Invent 2020: What's new in Amazon ElastiCache
- AWS re:Invent 2019: What's new in Amazon ElastiCache
- AWS re:Invent 2017: What's new in Amazon ElastiCache
- DAT204—Building Scalable Applications on AWS NoSQL Services (re:Invent 2015)
- DAT207—Accelerating Application Performance with Amazon ElastiCache (AWS re:Invent 2013)

AWS re:Invent 2020: What's new in Amazon ElastiCache

AWS re:Invent 2020: What's new in Amazon ElastiCache

AWS re:Invent 2019: What's new in Amazon ElastiCache

AWS re:Invent 2019: What's new in Amazon ElastiCache

AWS re:Invent 2017: What's new in Amazon ElastiCache

AWS re:Invent 2017: What's new in Amazon ElastiCache

DAT204—Building Scalable Applications on AWS NoSQL Services (re:Invent 2015)

En esta sesión, analizamos los beneficios de las bases de datos NoSQL y se realiza un recorrido por los principales servicios NoSQL que ofrece AWS: Amazon DynamoDB y Amazon ElastiCache.

A continuación, escuchamos a dos de nuestros principales clientes, Expedia y Mapbox, que hablan sobre sus casos de uso y desafíos en la arquitectura, y cómo los han abordado con los servicios NoSQL de AWS, incluidos patrones de diseño y prácticas recomendadas. Terminará esta sesión con una mejor comprensión de NoSQL y sus potentes capacidades, listo para abordar los desafíos de su base de datos con confianza.

DAT204—Building Scalable Applications on AWS NoSQL Services (re:Invent 2015)

DAT207—Accelerating Application Performance with Amazon ElastiCache (AWS re:Invent 2013)

En este video, aprenderá a utilizar Amazon ElastiCache para implementar fácilmente un sistema de almacenamiento de caché en memoria a fin de acelerar el rendimiento de su aplicación. Mostramos cómo utilizar Amazon ElastiCache para mejorar la latencia de su aplicación y reducir la carga de sus servidores de bases de datos. Asimismo, mostraremos cómo crear una capa de almacenamiento en caché que resulte fácil de administrar y escalar a medida que crezca su aplicación. Durante esta sesión, recorreremos diversas situaciones y casos de uso que pueden beneficiarlo al permitir el almacenamiento en caché y explicaremos las características que ofrece Amazon ElastiCache.

DAT207 - Accelerating Application Performance with Amazon ElastiCache (re:Invent 2013)

Videos avanzados

En los siguientes videos se tratan temas de Amazon ElastiCache más avanzados.

Temas

- Design for success with Amazon ElastiCache best practices (re:Invent 2020)
- Supercharge your real-time apps with Amazon ElastiCache (re:Invent 2019)
- Best practices: migrating Redis OSS clusters from Amazon EC2 to ElastiCache (re:Invent 2019)
- Scaling a Fantasy Sports Platform with Amazon ElastiCache & Amazon Aurora STP11 (re:Invent 2018)
- Reliable & Scalable Redis OSS in the Cloud with Amazon ElastiCache (re:Invent 2018)
- ElastiCache Deep Dive: Design Patterns for In-Memory Data Stores (re:Invent 2018)
- DAT305—Amazon ElastiCache Deep Dive (re:Invent 2017)
- DAT306—Amazon ElastiCache Deep Dive (re:Invent 2016)
- DAT317—How IFTTT uses ElastiCache for Redis OSS to Predict Events (re:Invent 2016)
- DAT407—Amazon ElastiCache Deep Dive (re:Invent 2015)

- SDD402—Amazon ElastiCache Deep Dive (re:Invent 2014)
- DAT307—Deep Dive into Amazon ElastiCache Architecture and Design Patterns (re:Invent 2013)

Design for success with Amazon ElastiCache best practices (re:Invent 2020)

Con el crecimiento explosivo de aplicaciones críticas para el negocio y en tiempo real creadas en Redis OSS, la disponibilidad, la escalabilidad y la seguridad se han convertido en las principales consideraciones. Conozca las prácticas recomendadas para configurar Amazon ElastiCache a fin de tener éxito con el escalado en línea, la alta disponibilidad en implementaciones Multi-AZ y las configuraciones de seguridad.

Design for success with Amazon ElastiCache best practices (re:Invent 2020)

Supercharge your real-time apps with Amazon ElastiCache (re:Invent 2019)

Con el rápido crecimiento de la adopción de la nube y los escenarios nuevos que habilita, las aplicaciones necesitan latencia de microsegundos y un alto rendimiento para soportar millones de solicitudes por segundo. Tradicionalmente, los desarrolladores han confiado en hardware especializado y soluciones alternativas, como bases de datos con base en disco combinadas con técnicas de reducción de datos, a fin de administrar datos para aplicaciones en tiempo real. Estos enfoques pueden ser costosos y no se pueden escalar. Descubra cómo puede aumentar el rendimiento de las aplicaciones en tiempo real mediante la utilización de Amazon ElastiCache completamente administrado y en memoria para obtener un rendimiento extremo, una alta escalabilidad, disponibilidad y seguridad.

Supercharge your real-time apps with Amazon ElastiCache (re:Invent 2019:)

Best practices: migrating Redis OSS clusters from Amazon EC2 to ElastiCache (re:Invent 2019)

Administrar clústeres de Redis OSS por su cuenta puede ser difícil. Debe aprovisionar hardware, aplicar parches de software, realizar copias de seguridad de los datos y monitorear las cargas de trabajo de forma constante. Ahora, con la nueva característica de migración en línea para Amazon ElastiCache, puede mover fácilmente sus datos de Redis OSS autoalojado en Amazon EC2 a Amazon ElastiCache completamente administrado, con el modo de clúster deshabilitado. En esta sesión, obtendrá información sobre la nueva herramienta de migración en línea, verá una demostración y, lo que es más importante, aprenderá prácticas recomendadas para lograr una migración sin problemas a Amazon ElastiCache.

Best practices: migrating Redis OSS clusters from Amazon EC2 to ElastiCache (re:Invent 2019)

Scaling a Fantasy Sports Platform with Amazon ElastiCache & Amazon Aurora STP11 (re:Invent 2018)

Dream11 es la empresa emergente líder en tecnología deportiva de la India. Tiene una base creciente de más de 40 millones de usuarios que practican varios deportes, incluido cricket de fantasía, fútbol y baloncesto, y actualmente asiste a un millón de usuarios simultáneos, que producen tres millones de solicitudes por minuto en un tiempo de respuesta de 50 milisegundos. En esta charla, el CTO de Dream11, Amit Sharma, explica cómo la empresa utiliza Amazon Aurora y Amazon ElastiCache para gestionar el tráfico rápido, que puede triplicarse en un periodo de respuesta de 30 segundos. Sharma también habla sobre la escalabilidad de las transacciones sin bloquear y comparte los pasos para gestionar el tráfico rápido, con lo que asiste a cinco millones de usuarios activos diarios. Título completo: AWS re:Invent 2018: Scaling a Fantasy Sports Platform with Amazon ElastiCache & Amazon Aurora (STP11)

Scaling a Fantasy Sports Platform with Amazon ElastiCache & Amazon Aurora STP11 (re:Invent 2018)

Reliable & Scalable Redis OSS in the Cloud with Amazon ElastiCache (re:Invent 2018)

Esta sesión cubre las características y mejoras de nuestro servicio compatible con Redis OSS, Amazon ElastiCache para Redis OSS. Cubrimos características clave, como Redis OSS 5, mejoras de escalabilidad y rendimiento, seguridad y conformidad normativa y mucho más. También analizamos las próximas características y los casos prácticos de clientes.

Reliable & Scalable Redis OSS in the Cloud with Amazon ElastiCache (re:Invent 2018)

ElastiCache Deep Dive: Design Patterns for In-Memory Data Stores (re:Invent 2018)

En esta sesión, ofrecemos un vistazo entre bastidores para aprender sobre el diseño y la arquitectura de Amazon ElastiCache. Consulte patrones de diseño comunes con nuestras ofertas de Memcached y Redis OSS, y conozca cómo los clientes los han utilizado para procesamiento de datos en memoria con el fin de reducir la latencia y mejorar el rendimiento de las aplicaciones. Analizamos las prácticas recomendadas, los patrones de diseño y los patrones no recomendados de ElastiCache.

ElastiCache Deep Dive: Design Patterns for In-Memory Data Stores (re:Invent 2018)

DAT305—Amazon ElastiCache Deep Dive (re:Invent 2017)

Profundice para obtener más información sobre el diseño y la arquitectura de Amazon ElastiCache. Consulte patrones de diseño comunes con nuestras ofertas de Memcached y Redis OSS y vea cómo los clientes los han utilizado para operaciones en memoria con el fin de reducir la latencia y mejorar

el rendimiento de las aplicaciones. Durante este video, analizaremos las prácticas recomendadas, los patrones de diseño y los patrones no recomendados de ElastiCache.

El video introduce lo siguiente:

- Redistribución en línea de ElastiCache para Redis OSS
- Cifrado y seguridad de ElastiCache
- ElastiCache para Redis OSS versión 3.2.10

DAT305—Amazon ElastiCache Deep Dive (re:Invent 2017)

DAT306—Amazon ElastiCache Deep Dive (re:Invent 2016)

Profundice para obtener más información sobre el diseño y la arquitectura de Amazon ElastiCache. Consulte patrones de diseño comunes con nuestras ofertas de Memcached y Redis OSS y vea cómo los clientes los han utilizado para operaciones en memoria con el fin de reducir la latencia y mejorar el rendimiento de las aplicaciones. Durante esta sesión, analizaremos las prácticas recomendadas, los patrones de diseño y los patrones no recomendados de ElastiCache.

DAT306—Amazon ElastiCache Deep Dive (re:Invent 2016)

DAT317—How IFTTT uses ElastiCache for Redis OSS to Predict Events (re:Invent 2016)

IFTTT es un servicio gratuito que permite a las personas hacer mucho más con sus servicios favoritos, desde automatizar tareas sencillas hasta transformar el modo en el que controlar su hogar e interactuar con este. IFTTT utiliza ElastiCache para Redis OSS para almacenar el historial de ejecución de transacciones y programar las predicciones, además de indexar documentos de registro en Amazon S3. Vea esta sesión para conocer la capacidad de scripting de Lua y descubrir cómo los tipos de datos de Redis OSS han permitido a las personas conseguir algo que de otra forma no habrían podido lograr.

DAT317—How IFTTT uses ElastiCache for Redis OSS to Predict Events (re:Invent 2016)

DAT407—Amazon ElastiCache Deep Dive (re:Invent 2015)

Profundice para obtener más información sobre el diseño y la arquitectura de Amazon ElastiCache. Consulte patrones de diseño comunes de nuestras ofertas de Memcached y Redis OSS y vea cómo los clientes los han utilizado para operaciones en memoria y han conseguido una latencia y un rendimiento mejorados para sus aplicaciones. Durante esta sesión, analizaremos las prácticas recomendadas, los patrones de diseño y los patrones no recomendados de Amazon ElastiCache.

DAT407—Amazon ElastiCache Deep Dive (re:Invent 2015)

SDD402—Amazon ElastiCache Deep Dive (re:Invent 2014)

En este vídeo, examinaremos casos de uso comunes del almacenamiento en caché, los motores ElastiCache para Memcached y ElastiCache para Redis OSS, patrones que le ayudan a determinar qué motor es mejor según sus necesidades, el uso coherente de hash y mucho más, como medio para crear aplicaciones rápidas y escalables. Frank Wiebe, científico principal de Adobe, detalla la forma en la que Adobe utiliza Amazon ElastiCache para mejorar la experiencia del cliente y escalar su negocio.

DAT402—Amazon ElastiCache Deep Dive (re:Invent 2014)

DAT307—Deep Dive into Amazon ElastiCache Architecture and Design Patterns (re:Invent 2013)

En este video, examinaremos el almacenamiento en caché, estrategias de almacenamiento en caché, escalado ascendente y monitoreo. También compararemos los motores de ElastiCache para Memcached y ElastiCache para Redis OSS. Asimismo, durante esta sesión, analizaremos las prácticas recomendadas y los patrones de diseño de Amazon ElastiCache.

DAT307 - Deep Dive into Amazon ElastiCache Architecture and Design Patterns (AWS re:Invent 2013).

Administración de nodos en ElastiCache

Un nodo es el bloque de creación más pequeño de una implementación de Amazon ElastiCache. Se trata de un fragmento de tamaño fijo de RAM segura conectada a la red. Cada nodo ejecuta el motor que se eligió cuando el clúster se creó o se modificó por última vez. Cada nodo tiene su propio puerto y nombre de servicio de nombres de dominio (DNS). Se admiten varios tipos de nodos de ElastiCache, cada uno de ellos con distintas cantidades de memoria y potencia informática asociadas.

Consulte para ver una explicación más detallada sobre el tamaño de nodo que debe utilizarse Elección del tamaño del nodo.

Por lo general, gracias a la compatibilidad con las particiones, las implementaciones de Valkey o Redis OSS (modo de clúster habilitado) tienen una serie de nodos más pequeños. En contraposición, las implementaciones de Valkey o Redis OSS (modo de clúster deshabilitado) tienen menos nodos de gran tamaño en un clúster. Consulte para ver una explicación más detallada sobre el tamaño de nodo que debe utilizarse Elección del tamaño del nodo.

Temas

- Visualización del estado de los nodos en ElastiCache
- Nodos y particiones de Valkey o Redis OSS
- Conexión a los nodos
- Tipos de nodos compatibles
- Reinicio de nodos
- Reemplazo de nodos (Valkey y Redis OSS)
- Reemplazo de nodos (Memcached)
- Nodos reservados
- Migración de nodos de generación anterior

Entre las opciones importantes en relación con los nodos, se encuentran las siguientes:

- Adición de nodos a un clúster de ElastiCache
- Eliminación de nodos de un clúster de ElastiCache
- Escalado de ElastiCache
- Búsqueda de puntos de conexión en ElastiCache
- Identificación automática de los nodos en el clúster (Memcached)

Visualización del estado de los nodos en ElastiCache

Con la consola de ElastiCache, puede acceder rápidamente al estado del nodo de ElastiCache. El estado de un nodo de ElastiCache indica la situación de este. Puede utilizar los siguientes procedimientos para ver el estado del nodo de ElastiCache en la consola Amazon ElastiCache, el comando de la AWS CLI o la operación API.

Los posibles valores de estado de los nodos de ElastiCache se muestran en la siguiente tabla. En esta tabla, también se muestra si se le facturará por el nodo de ElastiCache.

Tipo	Facturado	Descripción
`available`	Facturado	El nodo de ElastiCache funciona correctamente y está disponible.

Tipo	Facturado	Descripción
`creating`	No facturado	Se está creando el nodo de ElastiCache. No se puede obtener acceso al nodo mientras se está creando.
`deleting`	No facturado	Se está eliminando el nodo de ElastiCache.
`modifying`	Facturado	El nodo de ElastiCache se está modificando porque un cliente ha solicitado la modificación.

Tipo	Facturado	Descripción
`updating`	Facturado	El estado Actualizando indica que se cumple una o varias de las siguientes condiciones en el nodo de Amazon ElastiCache: • Se está aplicando el parche al nodo de ElastiCache como parte de la actualización del servicio. Para obtener más información sobre las actualizaciones de servicio, consulte la Página de ayuda de actualizaciones de servicio y mantenimiento administrado de Amazon ElastiCache. • Los grupos de seguridad de VPC se están actualizando para el clúster de ElastiCache. • El clúster de ElastiCache se está escalando verticalmente o reduciendo verticalmente. • Se están modificando las configuraciones de entrega de registros para el clúster de ElastiCache. • Hay pendiente una operación de eliminación para el nodo de ElastiCache.

Tipo	Facturado	Descripción
		• La contraseña de ElastiCache se está actualizando o rotando con AWS Secrets Manager.
`rebooting cluster nodes`	Facturado	El nodo de base de datos se está reiniciando por la solicitud de un cliente o porque un proceso de Amazon ElastiCache lo requiere.
`incompatible_parameters`	No facturado	Amazon ElastiCache no puede iniciar el nodo porque los parámetros especificados en el grupo de parámetros del nodo no son compatibles con el nodo. Revierta los cambios de los parámetros o hágalos compatibles con el nodo para recuperar el acceso a este. Para obtener más información acerca de los parámetros incompatibles, consulte la lista Eventos para el nodo de ElastiCache.

Tipo	Facturado	Descripción
`incompatible_network`	No facturado	El estado de red incompatible indica que se cumple una o varias de las siguientes condiciones en el nodo de Amazon ElastiCache: • No hay direcciones IP disponibles en la subred en la que se lanzó el nodo de ElastiCache. • La subred mencionada en el grupo de subredes de ElastiCache ya no existe en Amazon Virtual Private Cloud (Amazon VPC).

Tipo	Facturado	Descripción
`restore_failed`	No facturado	El estado de error en la restauración indica que se cumple una o varias de las siguientes condiciones en el nodo de Amazon ElastiCache: • Se produjo un error al reemplazar el nodo debido a un estado repetido de capacidad insuficiente de la instancia. Esto suele ocurrir cuando se ejecutan nodos de generaciones anteriores que están al final de su vida útil. Sin embargo, también puede ocurrir al reemplaza r los nodos de la generació n actual cuando AWS no tiene suficiente capacidad bajo demanda para atender su solicitud en la zona de disponibilidad especific ada. Para obtener más información sobre cómo reparar o eliminar estos nodos, consulte Migración de nodos de generación anterior. • No se pudo restaurar la instantánea de RDB especificada. • Se suspendió la cuenta de AWS del clúster de ElastiCache.

Tipo	Facturado	Descripción
		• Se produjo un error en el nodo y no se pudo recuperar.
`snapshotting`	Facturado	ElastiCache está creando una instantánea del nodo de Valkey o Redis OSS.

Visualización del estado de los nodos de ElastiCache con la consola

Para ver el estado de un nodo de ElastiCache con la consola:

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Clústeres de Redis OSS o Clústeres de Memcached. Aparecerá la página Cachés con la lista de nodos de ElastiCache. Para cada nodo, se muestra el valor del estado.
3. A continuación, puede ir a la pestaña Actualizaciones de servicio de la caché para ver la lista de actualizaciones de servicio aplicables a la caché.

Visualización del estado de los nodos de ElastiCache con la AWS CLI

Para ver el nodo de ElastiCache y su información de estado mediante la AWS CLI, utilice el comando `describe-cache-cluster`. Por ejemplo, el siguiente comando AWS CLI muestra cada nodo de ElastiCache.

```
aws elasticache describe-cache-clusters
```

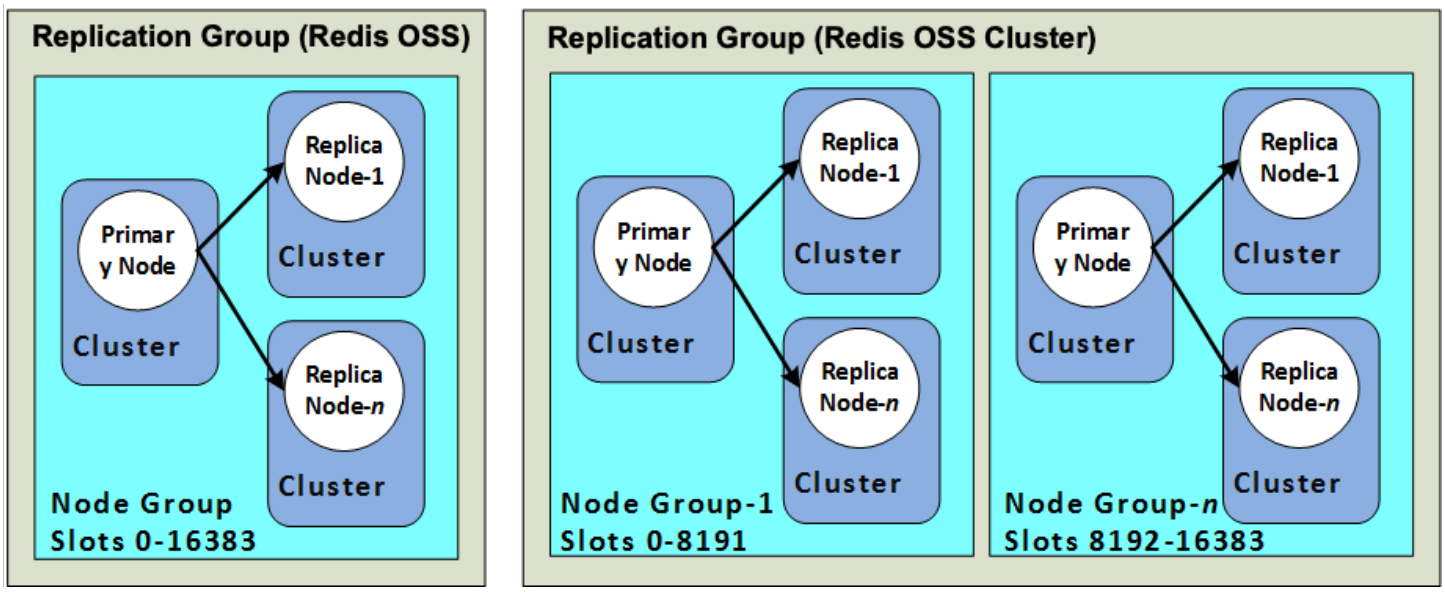
Visualización del estado de los nodos de ElastiCache con la API

Para ver el estado de los nodos de ElastiCache mediante la API de Amazon ElastiCache, llame a `DescribeCacheClusteroperation` con el indicador `ShowCacheNodeInfo` para recuperar información sobre nodos de caché específicos.

Nodos y particiones de Valkey o Redis OSS

Una partición (en la API y la CLI, un grupo de nodos) es un grupo de nodos clasificados de forma jerárquica en un clúster. Las particiones son compatibles con la reproducción. En una partición, un nodo funciona como nodo principal de lectura/escritura. Todos los demás nodos de la partición funcionan como réplicas de solo lectura del nodo principal. Valkey o Redis OSS versión 3.2 y posteriores admiten varias particiones en un clúster (en la API y la CLI, un grupo de replicación). Esta compatibilidad permite la partición de los datos en un clúster de Valkey o Redis OSS (modo de clúster habilitado).

El siguiente diagrama ilustra las diferencias entre un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) y un clúster de Valkey o Redis OSS (modo de clúster habilitado).



Los clústeres de Valkey o Redis OSS (modo de clúster habilitado) admiten la replicación a través de particiones. La operación de la API DescribeReplicationGroups (CLI: describe-replication-groups) muestra los grupos de nodos con los nodos miembros y el rol del nodo en el grupo de nodos, entre otros datos.

Al crear un clúster de Valkey o Redis OSS, se especifica si desea crear un clúster con la agrupación en clúster habilitada. Los clústeres de Valkey o Redis OSS (modo de clúster deshabilitado) nunca tienen más de una partición, que puede escalarse horizontalmente al añadir (hasta un total de cinco) o eliminar nodos de réplica de lectura. Para obtener más información, consulte Alta disponibilidad a través de grupos de reproducción, Adición de una réplica de lectura para Valkey o Redis OSS (modo de clúster deshabilitado) o Eliminación de una réplica de lectura para Valkey o Redis OSS (modo de clúster deshabilitado). Los clústeres de Valkey o Redis OSS (modo de clúster deshabilitado) también

pueden escalarse verticalmente al cambiar los tipos de nodos. Para obtener más información, consulte Escalado de nodos de réplica para Valkey o Redis OSS (modo de clúster deshabilitado).

El límite de nodos o particiones se puede aumentar a un máximo de 500 por clúster si la versión del motor de Valkey o Redis OSS es 5.0.6 o posteriores. Por ejemplo, puede elegir configurar un clúster de 500 nodos que oscila entre 83 particiones (uno primario y 5 réplicas por partición) y 500 particiones (único primario y sin réplicas). Asegúrese de que hay suficientes direcciones IP disponibles para acomodar el aumento. Algunos problemas comunes incluyen que las subredes del grupo de subredes tienen un rango CIDR demasiado pequeño o que otros clústeres comparten y utilizan considerablemente las subredes. Para obtener más información, consulte Creación de un grupo de subredes.

Para las versiones inferiores a 5.0.6, el límite es de 250 por clúster.

Para solicitar un aumento del límite, consulte AWS Service Limits y elija el tipo de límite Nodes per cluster per instance type (Nodos por clúster por tipo de instancias).

Después de crear un clúster de Valkey o Redis OSS (modo de clúster habilitado), se puede modificar (reducir horizontalmente o escalar verticalmente). Para obtener más información, consulte Escalado de ElastiCache y Reemplazo de nodos (Valkey y Redis OSS).

Cuando cree un clúster nuevo, puede inicializarlo con datos del clúster anterior para que no comience vacío. El enfoque anterior solo funciona si el grupo de clústeres tiene el mismo número de particiones que el clúster anterior. Esto puede resultar útil si necesita cambiar el tipo de nodo o la versión del motor. Para obtener más información, consulte Copias de seguridad manuales y Restauración desde una copia de seguridad a una nueva caché.

Conexión a los nodos

Conexión a nodos de Valkey o Redis OSS

Antes de intentar conectarse a los nodos de su clúster de Valkey o Redis OSS, debe disponer de los puntos de conexión para los nodos. Para encontrar los puntos de enlace, consulte lo siguiente:

- Búsqueda de puntos de conexión de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) (consola)
- Búsqueda de puntos de conexión para un clúster de Valkey o Redis OSS (modo de clúster habilitado) (consola)
- Búsqueda de puntos de conexión (AWS CLI)
- Búsqueda de puntos de enlace (API de ElastiCache)

En el ejemplo siguiente, se emplea la utilidad valkey-cli para conectarse a un clúster que ejecuta Valkey o Redis OSS.

> **Note**
>
> Para obtener más información sobre los comandos disponibles, consulte la página web Comandos.

Conexión a un clúster de Valkey o Redis OSS mediante valkey-cli

1. Conéctese a la instancia de Amazon EC2 mediante la utilidad de conexión que prefiera.

   > **Note**
   >
   > Para obtener instrucciones sobre cómo conectarse a una instancia de Amazon EC2, consulte la Guía de introducción de Amazon EC2.

2. Para la creación de `valkey-cli`, descargue e instale la recopilación del compilador GNU (`gcc`). En el símbolo del sistema de su instancia EC2, escriba el siguiente comando y especifique y en el mensaje de confirmación.

   ```
   sudo yum install gcc
   ```

Aparecerá una salida similar a la siguiente.

```
Loaded plugins: priorities, security, update-motd, upgrade-helper
Setting up Install Process
Resolving Dependencies
--> Running transaction check

...(output omitted)...

Total download size: 27 M
Installed size: 53 M
Is this ok [y/N]: y
Downloading Packages:
(1/11): binutils-2.22.52.0.1-10.36.amzn1.x86_64.rpm      | 5.2 MB     00:00
(2/11): cpp46-4.6.3-2.67.amzn1.x86_64.rpm                | 4.8 MB     00:00
(3/11): gcc-4.6.3-3.10.amzn1.noarch.rpm                  | 2.8 kB     00:00

...(output omitted)...

Complete!
```

3. Descargue y compile la utilidad valkey-cli. Esta utilidad se incluye en la distribución de software de Valkey. En el símbolo del sistema de su instancia EC2, escriba los comandos siguientes:

> **Note**
>
> En sistemas Ubuntu, antes de ejecutar `make`, ejecute `make distclean`.

```
wget https://github.com/valkey-io/valkey/archive/refs/tags/8.0.0.tar.gz
tar xvzf valkey-8.0.0.tar.gz
cd valkey-8.0.0
make distclean      # ubuntu systems only
make
```

4. En el símbolo del sistema de la instancia EC2, escriba los comandos siguientes:

```
src/valkey-cli -c -h mycachecluster.eaogs8.0001.usw2.cache.amazonaws.com -p 6379
```

Aparecerá un símbolo del sistema de Valkey o Redis OSS similar al siguiente.

```
redis mycachecluster.eaogs8.0001.usw2.cache.amazonaws.com 6379>
```

5. Pruebe la conexión mediante la ejecución de comandos de Valkey o Redis OSS.

   Ya se ha conectado al clúster y puede ejecutar comandos de Valkey o Redis OSS. A continuación, se muestran algunos comandos de ejemplo con sus respuestas de Valkey o Redis OSS.

```
set a "hello"          // Set key "a" with a string value and no expiration
OK
get a                  // Get value for key "a"
"hello"
get b                  // Get value for key "b" results in miss
(nil)
set b "Good-bye" EX 5  // Set key "b" with a string value and a 5 second expiration
get b
"Good-bye"
                    // wait 5 seconds
get b
(nil)                  // key has expired, nothing returned
quit                   // Exit from valkey-cli
```

Para conectarse a nodos o clústeres que tienen cifrado de capa de conexión segura (SSL) (en tránsito habilitado), consulte Cifrado en tránsito (TLS) de ElastiCache.

Conexión a los nodos de Memcached

Antes de intentar conectarse a su clúster de Memcached, debe disponer de los puntos de enlace para los nodos. Para encontrar los puntos de enlace, consulte lo siguiente:

- Búsqueda de puntos de conexión de un clúster (consola) (Memcached)
- Búsqueda de puntos de conexión (AWS CLI)
- Búsqueda de puntos de enlace (API de ElastiCache)

En el ejemplo siguiente, utilice la utilidad telnet para conectarse a un nodo que ejecuta Memcached.

> **Note**
>
> Para obtener más información acerca de Memcached y los comandos de Memcached disponibles, consulte el sitio web de Memcached.

Para conectarse a un nodo mediante telnet

1. Conéctese a la instancia de Amazon EC2 mediante el uso de la utilidad de conexión que prefiera.

   > **Note**
   >
   > Para obtener instrucciones sobre cómo conectarse a una instancia de Amazon EC2, consulte la Guía de introducción de Amazon EC2.

2. Descargue e instale la utilidad telnet en la instancia de Amazon EC2. En el símbolo del sistema de la instancia de Amazon EC2, escriba el siguiente comando y escriba y en el símbolo del sistema.

   ```
   sudo yum install telnet
   ```

   Aparecerá una salida similar a la siguiente.

   ```
   Loaded plugins: priorities, security, update-motd, upgrade-helper
   Setting up Install Process
   Resolving Dependencies
   --> Running transaction check

   ...(output omitted)...

   Total download size: 63 k
   Installed size: 109 k
   Is this ok [y/N]: y
   Downloading Packages:
   telnet-0.17-47.7.amzn1.x86_64.rpm                          |  63 kB     00:00

   ...(output omitted)...
   ```

```
Complete!
```

3. En el símbolo del sistema de la instancia de Amazon EC2, escriba el siguiente comando, sustituya el punto de enlace del nodo por el que se muestra en este ejemplo.

```
telnet mycachecluster.eaogs8.0001.usw2.cache.amazonaws.com 11211
```

Aparecerá una salida similar a la siguiente.

```
Trying 128.0.0.1...
Connected to mycachecluster.eaogs8.0001.usw2.cache.amazonaws.com.
Escape character is '^]'.
>
```

4. Pruebe la conexión mediante la ejecución de comandos de Memcached.

Ahora está conectado a un nodo y puede ejecutar comandos de Memcached. A continuación se muestra un ejemplo.

```
set a 0 0 5      // Set key "a" with no expiration and 5 byte value
hello            // Set value as "hello"
STORED
get a            // Get value for key "a"
VALUE a 0 5
hello
END
get b            // Get value for key "b" results in miss
END
>
```

Tipos de nodos compatibles

ElastiCache admite los siguientes tipos de nodos. En general, los tipos de la generación actual ofrecen más capacidad de memoria y potencia de computación a menor costo comparados con sus equivalentes de generaciones previas.

Para obtener más información sobre los detalles del rendimiento de cada tipo de nodo, consulte Tipos de instancia de Amazon EC2.

> **Note**
>
> Se admiten los siguientes tipos de instancias en las regiones de AWS Asia-Pacífico (Tailandia) y México (centro):
>
> - m7g/r7g: grande, xl, 2xl, 4xl, 8xl, 12xl y 16xl.
> - t3/t4g: micro, pequeño y mediano.

Para obtener información sobre el tamaño de nodo que debe utilizarse, consulte Elección del tamaño del nodo.

Temas

- Versión actual (Memcached)
- Generación actual (Valkey o Redis OSS)
- Tipos de nodos admitidos por región de AWS
- Instancias de rendimiento ampliable
- Información relacionada

Versión actual (Memcached)

Las siguientes tablas muestran el ancho de banda de base y de ampliación de los tipos de instancias que utilizan el mecanismo de créditos de E/S de red para superar el ancho de banda de base.

> **Note**
>
> Los tipos de instancias con rendimiento de red ampliable utilizan un mecanismo de créditos de E/S de red para superar el ancho de banda de base en función del esfuerzo.

General

Tipo de instancia	Versión mínima de Memcached compatible	Banda ancha de base (Gbps)	Banda ancha con ráfagas (Gbps)
cache.m7g.large		0,937	12,5
cache.m7g.xlarge		1,876	12,5
cache.m7g.2xlarge		3.75	15
cache.m7g.4xlarge		7.5	15
cache.m7g.8xlarge		15	N/A
cache.m7g.12xlarge		22,5	N/A
cache.m7g.16xlarge		30	N/A
cache.m6g.large	1.5.16	0.75	10.0
cache.m6g.xlarge	1.5.16	1,25	10.0
cache.m6g.2xlarge	1.5.16	2,5	10.0
cache.m6g.4xlarge	1.5.16	5.0	10.0
cache.m6g.8xlarge	1.5.16	12	N/A
cache.m6g.12xlarge	1.5.16	20	N/A
cache.m6g.16xlarge	1.5.16	25	N/A
cache.m5.large	1.5.16	0.75	10.0
cache.m5.xlarge	1.5.16	1,25	10.0
cache.m5.2xlarge	1.5.16	2,5	10.0
cache.m5.4xlarge	1.5.16	5.0	10.0

Tipo de instancia	Versión mínima de Memcached compatible	Banda ancha de base (Gbps)	Banda ancha con ráfagas (Gbps)
cache.m5.12xlarge	1.5.16	N/A	N/A
cache.m5.24xlarge	1.5.16	N/A	N/A
cache.m4.large	1.5.16	0,45	1.2
cache.m4.xlarge	1.5.16	0.75	2,8
cache.m4.2xlarge	1.5.16	1.0	10.0
cache.m4.4xlarge	1.5.16	2.0	10.0
cache.m4.10xlarge	1.5.16	5.0	10.0
cache.t4g.micro	1.5.16	0,064	5.0
cache.t4g.small	1.5.16	0,128	5.0
cache.t4g.medium	1.5.16	0,256	5.0
cache.t3.micro	1.5.16	0,064	5.0
cache.t3.small	1.5.16	0,128	5.0
cache.t3.medium	1.5.16	0,256	5.0
cache.t2.micro	1.5.16	0,064	1,024
cache.t2.small	1.5.16	0,128	1,024
cache.t2.medium	1.5.16	0,256	1,024

Memoria optimizada para Memcached

Tipo de instancia	Versión mínima compatible	Banda ancha de base (Gbps)	Banda ancha con ráfagas (Gbps)
cache.r7g.large		0,937	12,5
cache.r7g.xlarge		1,876	12,5
cache.r7g.2xlarge		3.75	15
cache.r7g.4xlarge		7.5	15
cache.r7g.8xlarge		15	N/A
cache.r7g.12xlarge		22,5	N/A
cache.r7g.16xlarge		30	N/A
cache.r6g.large	1.5.16	0.75	10.0
cache.r6g.xlarge	1.5.16	1,25	10.0
cache.r6g.2xlarge	1.5.16	2,5	10.0
cache.r6g.4xlarge	1.5.16	5.0	10.0
cache.r6g.8xlarge	1.5.16	12	N/A
cache.r6g.12xlarge	1.5.16	20	N/A
cache.r6g.16xlarge	1.5.16	25	N/A
cache.r5.large	1.5.16	0.75	10.0
cache.r5.xlarge	1.5.16	1,25	10.0
cache.r5.2xlarge	1.5.16	2,5	10.0
cache.r5.4xlarge	1.5.16	5.0	10.0
cache.r5.12xlarge	1.5.16	20	N/A
cache.r5.24xlarge	1.5.16	25	N/A

Tipo de instancia	Versión mínima compatible	Banda ancha de base (Gbps)	Banda ancha con ráfagas (Gbps)
cache.r4.large	1.5.16	0.75	10.0
cache.r4.xlarge	1.5.16	1,25	10.0
cache.r4.2xlarge	1.5.16	2,5	10.0
cache.r4.4xlarge	1.5.16	5.0	10.0
cache.r4.8xlarge	1.5.16	12	N/A
cache.r4.16xlarge	1.5.16	25	N/A

Red optimizada para Memcached

Tipo de instancia	Versión mínima compatible	Banda ancha de base (Gbps)	Banda ancha con ráfagas (Gbps)
cache.c7gn.large	1.6.6	6,25	30
cache.c7gn.xlarge	1.6.6	12,5	40
cache.c7gn.2xlarge	1.6.6	25	50
cache.c7gn.4xlarge	1.6.6	50	N/A
cache.c7gn.8xlarge	1.6.6	100	N/A
cache.c7gn.12xlarge	1.6.6	150	N/A
cache.c7gn.16xlarge	1.6.6	200	N/A

Generación actual (Valkey o Redis OSS)

Para obtener más información sobre la generación anterior, consulte Nodos de la generación anterior.

> **Note**
>
> Los tipos de instancias con rendimiento de red ampliable utilizan un mecanismo de créditos de E/S de red para superar el ancho de banda de base en función del esfuerzo.

General

Tipo de instancia	Versión mínima de Redis OSS compatible	E/S mejorada con Redis OSS 5.0.6 y versiones posteriores	Descarga de TLS con Redis OSS 6.2.5 y versiones posteriores	Multiplexación de E/S mejorada con Redis OSS 7.0 y versiones posteriores	Banda ancha de base (Gbps)	Banda ancha con ráfagas (Gbps)
cache.m7g.large	6.2	N	N	N	0,937	12,5
cache.m7g.xlarge	6.2	S	S	S	1,876	12,5
cache.m7g.2xlarge	6.2	S	S	S	3.75	15
cache.m7g.4xlarge	6.2	S	S	S	7.5	15
cache.m7g.8xlarge	6.2	S	S	S	15	N/A
cache.m7g.12xlarge	6.2	S	S	S	22,5	N/A

Tipo de instancia	Versión mínima de Redis OSS compatible	E/S mejorada con Redis OSS 5.0.6 y versiones posteriores	Descarga de TLS con Redis OSS 6.2.5 y versiones posteriores	Multiplexación de E/S mejorada con Redis OSS 7.0 y versiones posteriores	Banda ancha de base (Gbps)	Banda ancha con ráfagas (Gbps)
cache.m7g.16xlarge	6.2	S	S	S	30	N/A
cache.m6g.large	5.0.6	N	N	N	0.75	10.0
cache.m6g.xlarge	5.0.6	S	S	S	1,25	10.0
cache.m6g.2xlarge	5.0.6	S	S	S	2,5	10.0
cache.m6g.4xlarge	5.0.6	S	S	S	5.0	10.0
cache.m6g.8xlarge	5.0.6	S	S	S	12	N/A
cache.m6g.12xlarge	5.0.6	S	S	S	20	N/A
cache.m6g.16xlarge	5.0.6	S	S	S	25	N/A
cache.m5.large	3.2.4	N	N	N	0.75	10.0
cache.m5.xlarge	3.2.4	S	N	N	1,25	10.0

Tipo de instancia	Versión mínima de Redis OSS compatible	E/S mejorada con Redis OSS 5.0.6 y versiones posteriores	Descarga de TLS con Redis OSS 6.2.5 y versiones posteriores	Multiplexación de E/S mejorada con Redis OSS 7.0 y versiones posteriores	Banda ancha de base (Gbps)	Banda ancha con ráfagas (Gbps)
cache.m5.2xlarge	3.2.4	S	S	S	2,5	10.0
cache.m5.4xlarge	3.2.4	S	S	S	5.0	10.0
cache.m5.12xlarge	3.2.4	S	S	S	12	N/A
cache.m5.24xlarge	3.2.4	S	S	S	25	N/A
cache.m4.large	3.2.4	N	N	N	0,45	1.2
cache.m4.xlarge	3.2.4	S	N	N	0.75	2,8
cache.m4.2xlarge	3.2.4	S	S	S	1.0	10.0
cache.m4.4xlarge	3.2.4	S	S	S	2.0	10.0
cache.m4.10xlarge	3.2.4	S	S	S	5.0	10.0
cache.t4g.micro	3.2.4	N	N	N	0,064	5.0

Tipo de instancia	Versión mínima de Redis OSS compatible	E/S mejorada con Redis OSS 5.0.6 y versiones posteriores	Descarga de TLS con Redis OSS 6.2.5 y versiones posteriores	Multiplexación de E/S mejorada con Redis OSS 7.0 y versiones posteriores	Banda ancha de base (Gbps)	Banda ancha con ráfagas (Gbps)
cache.t4g.small	5.0.6	N	N	N	0,128	5.0
cache.t4g.medium	5.0.6	N	N	N	0,256	5.0
cache.t3.micro	3.2.4	N	N	N	0,064	5.0
cache.t3.small	3.2.4	N	N	N	0,128	5.0
cache.t3.medium	3.2.4	N	N	N	0,256	5.0
cache.t2.micro	3.2.4	N	N	N	0,064	1,024
cache.t2.small	3.2.4	N	N	N	0,128	1,024
cache.t2.medium	3.2.4	N	N	N	0,256	1,024

Optimizadas para la memoria

Tipo de instancia	Versión mínima de Redis OSS compatible	E/S mejorada con Redis OSS 5.0.6 y versiones posteriores	Descarga de TLS con Redis OSS 6.2.5 y versiones posteriores	Multiplexación de E/S mejorada con Redis OSS 7.0 y versiones posteriores	Banda ancha de base (Gbps)	Banda ancha con ráfagas (Gbps)
cache.r7g.large	6.2	N	N	N	0,937	12,5
cache.r7g.xlarge	6.2	S	S	S	1,876	12,5
cache.r7g.2xlarge	6.2	S	S	S	3.75	15
cache.r7g.4xlarge	6.2	S	S	S	7.5	15
cache.r7g.8xlarge	6.2	S	S	S	15	N/A
cache.r7g.12xlarge	6.2	S	S	S	22,5	N/A
cache.r7g.16xlarge	6.2	S	S	S	30	N/A
cache.r6g.large	5.0.6	N	N	N	0.75	10.0
cache.r6g.xlarge	5.0.6	S	S	S	1,25	10.0
cache.r6g.2xlarge	5.0.6	S	S	S	2,5	10.0

Tipo de instancia	Versión mínima de Redis OSS compatible	E/S mejorada con Redis OSS 5.0.6 y versiones posteriores	Descarga de TLS con Redis OSS 6.2.5 y versiones posteriores	Multiplexación de E/S mejorada con Redis OSS 7.0 y versiones posteriores	Banda ancha de base (Gbps)	Banda ancha con ráfagas (Gbps)
cache.r6g.4xlarge	5.0.6	S	S	S	5.0	10.0
cache.r6g.8xlarge	5.0.6	S	S	S	12	N/A
cache.r6g.12xlarge	5.0.6	S	S	S	20	N/A
cache.r6g.16xlarge	5.0.6	S	S	S	25	N/A
cache.r5.large	3.2.4	N	N	N	0.75	10.0
cache.r5.xlarge	3.2.4	S	N	N	1,25	10.0
cache.r5.2xlarge	3.2.4	S	S	S	2,5	10.0
cache.r5.4xlarge	3.2.4	S	S	S	5.0	10.0
cache.r5.12xlarge	3.2.4	S	S	S	12	N/A
cache.r5.24xlarge	3.2.4	S	S	S	25	N/A

Tipo de instancia	Versión mínima de Redis OSS compatible	E/S mejorada con Redis OSS 5.0.6 y versiones posteriores	Descarga de TLS con Redis OSS 6.2.5 y versiones posteriores	Multiplexación de E/S mejorada con Redis OSS 7.0 y versiones posteriores	Banda ancha de base (Gbps)	Banda ancha con ráfagas (Gbps)
cache.r4.large	3.2.4	N	N	N	0.75	10.0
cache.r4.xlarge	3.2.4	S	N	N	1,25	10.0
cache.r4.2xlarge	3.2.4	S	S	S	2,5	10.0
cache.r4.4xlarge	3.2.4	S	S	S	5.0	10.0
cache.r4.8xlarge	3.2.4	S	S	S	12	N/A
cache.r4.16xlarge	3.2.4	S	S	S	25	N/A

Memoria optimizada con la organización de datos en niveles

Tipo de instancia	Versión mínima de Redis OSS compatible	E/S mejorada con Redis OSS 5.0.6 y versiones posteriores	Descarga de TLS con Redis OSS 6.2.5 y versiones posteriores	Multiplexación de E/S mejorada con Redis OSS 7.0 y versiones posteriores	Banda ancha de base (Gbps)	Banda ancha con ráfagas (Gbps)
cache.r6gd.xlarge	6.2.0	S	N	N	1,25	10
cache.r6gd.2xlarge	6.2.0	S	S	S	2,5	10
cache.r6gd.4xlarge	6.2.0	S	S	S	5.0	10
cache.r6gd.8xlarge	6.2.0	S	S	S	12	N/A
cache.r6gd.12xlarge	6.2.0	S	S	S	20	N/A
cache.r6gd.16xlarge	6.2.0	S	S	S	25	N/A

Optimizado para red

Tipo de instancia	Versión mínima de Redis OSS compatible	E/S mejorada con Redis OSS 5.0.6 y versiones posteriores	Descarga de TLS con Redis OSS 6.2.5 y versiones posteriores	Multiplexación de E/S mejorada con Redis OSS 7.0 y versiones posteriores	Banda ancha de base (Gbps)	Banda ancha con ráfagas (Gbps)
cache.c7gn.large	6.2	N	N	N	6,25	30
cache.c7gn.xlarge	6.2	S	S	S	12,5	40
cache.c7gn.2xlarge	6.2	S	S	S	25	50
cache.c7gn.4xlarge	6.2	S	S	S	50	N/A
cache.c7gn.8xlarge	6.2	S	S	S	100	N/A
cache.c7gn.12xlarge	6.2	S	S	S	150	N/A
cache.c7gn.16xlarge	6.2	S	S	S	200	N/A

Tipos de nodos admitidos por región de AWS

Los tipos de nodos admitidos pueden variar entre regiones de AWS. Para obtener más información, consulte Precios de Amazon ElastiCache.

Instancias de rendimiento ampliable

Puede lanzar nodos de caché T4g, T3 estándar y T2 estándar ampliables de uso general en Amazon ElastiCache. Estos nodos proporcionan un nivel básico de rendimiento de la CPU con la capacidad de explotar el uso de la CPU en cualquier momento hasta que se agoten los créditos acumulados. Un crédito de CPU proporciona el rendimiento de un núcleo de CPU completo durante un minuto.

Los nodos T4g, T3 y T2 de Amazon ElastiCache se han configurado como estándar y son adecuados para cargas de trabajo con un uso de la CPU promedio situado siempre por debajo del rendimiento de referencia de la instancia. Para ampliar por encima de la referencia, la instancia gasta créditos que ha acumulado en su saldo de créditos de CPU. Si el nodo se queda sin créditos acumulados, el rendimiento se reduce gradualmente hasta el nivel de rendimiento de referencia. Esta disminución gradual garantiza que el nodo no experimente una caída brusca del rendimiento cuando se agote el saldo de créditos de CPU acumulado. Para obtener más información, consulte Créditos de CPU y rendimiento de referencia para las instancias de rendimiento ampliable en la Guía del usuario de Amazon EC2.

En la siguiente tabla, se enumeran los tipos de nodos de rendimiento ampliables y la velocidad a la que se obtienen los créditos de CPU por hora. También se muestra el número máximo de créditos de CPU obtenidos que un nodo puede acumular y el número de vCPU por nodo. Además, se proporciona el nivel de rendimiento de referencia como porcentaje de un rendimiento básico completo (mediante una única vCPU).

Créditos de CPU obtenidos por hora	Créditos máximos ganados que se pueden acumular*	vCPU	Rendimiento de referencia por vCPU	Memoria (GiB)	Rendimiento de la red
12	288	2	10%	0,5	Hasta 5 gigabits
24	576	2	20%	1,37	Hasta 5 gigabits
24	576	2	20%	3,09	Hasta 5 gigabits

Créditos de CPU obtenidos por hora	Créditos máximos ganados que se pueden acumular*	vCPU	Rendimiento de referencia por vCPU	Memoria (GiB)	Rendimiento de la red
12	288	2	10%	0,5	Hasta 5 gigabits
24	576	2	20%	1,37	Hasta 5 gigabits
24	576	2	20%	3,09	Hasta 5 gigabits
6	144	1	10%	0,5	Baja a moderada
12	288	1	20%	1,55	Baja a moderada
24	576	2	20%	3,22	Baja a moderada

* El número de créditos que se pueden acumular es equivalente a la cantidad de créditos que se pueden obtener en un periodo de 24 horas.

** El rendimiento de referencia de la tabla es por vCPU. Algunos tamaños de nodos que tienen más de una vCPU. Para estos, calcule la utilización de CPU de referencia correspondiente al nodo con la multiplicación del porcentaje de vCPU por el número de vCPU.

Las siguientes métricas de crédito de CPU se encuentran disponibles para las instancias de rendimiento ampliable T3 y T4g:

> **Note**
>
> Estas métricas no se encuentran disponibles para las instancias de rendimiento ampliable T2.

- `CPUCreditUsage`
- `CPUCreditBalance`

Para obtener más información sobre estas métricas, consulte Métricas de crédito de CPU.

Además, tenga en cuenta estos detalles:

- Todos los tipos de nodos de generación actuales se crean en una Virtual Private Cloud (VPC) basada en Amazon VPC de forma predeterminada.
- Los archivos de Redis OSS solo anexados (AOF) no se admiten en las instancias T2. Las variables de configuración de Redis OSS `appendonly` y `appendfsync` no se admiten en la versión 2.8.22 de Redis OSS y posteriores.

Información relacionada

- Características y detalles del producto Amazon ElastiCache
- Parámetros específicos de tipo de nodo para Memcached
- Parámetros de Valkey y Redis OSS
- Cifrado en tránsito (TLS)

Reinicio de nodos

Algunos cambios requieren que el clúster de Valkey, Memcached o Redis OSS se reinicie para que los cambios se apliquen. Por ejemplo, en el caso de algunos parámetros, el cambio del valor del parámetro en un grupo de parámetros solo se aplica después de reiniciar.

Temas

- Reinicio de nodos de Redis OSS (solo modo de clúster deshabilitado)
- Reinicio de un clúster para Memcached

Reinicio de nodos de Redis OSS (solo modo de clúster deshabilitado)

En el caso de los clústeres de Valkey o Redis OSS (modo de clúster deshabilitado), los parámetros de los grupos de parámetros que se aplican únicamente después del reinicio son:

- activerehashing
- bases de datos

Los nodos de Redis solo se pueden actualizar a través de la consola de ElastiCache. Solo puede reiniciar un solo nodo a la vez. Si desea reiniciar varios nodos, debe repetir el proceso para cada uno de ellos.

Cambios de parámetros de Valkey o Redis OSS (modo de clúster habilitado)

Si realiza cambios en los siguientes parámetros en un clúster de Valkey o Redis OSS (modo de clúster habilitado), realice los siguientes pasos.

- activerehashing
- bases de datos

1. Cree una copia de seguridad manual de su clúster. Consulte Copias de seguridad manuales.
2. Elimine el clúster de Valkey o Redis OSS (modo de clúster habilitado). Consulte Eliminación de un clúster en ElastiCache.
3. Restaure el clúster con la copia de seguridad y el grupo de parámetros modificados para propagar los datos en el nuevo clúster. Consulte Restauración desde una copia de seguridad a una nueva caché.

Los cambios en otros parámetros no requieren esta acción.

Utilización de la Consola de administración de AWS

Es posible reiniciar un nodo con la consola de ElastiCache.

Para reiniciar un nodo (consola)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. Desde la lista en la esquina superior derecha, elija la región de AWS que corresponda.
3. En el panel de navegación izquierdo, elija Redis OSS.

   Aparecerá una lista de clústeres que ejecutan Redis OSS.
4. Elija el clúster en Cluster Name (Nombre de clúster).
5. En Node name (Nombre de nodo), elija el botón de opción situado junto al nodo que desea reiniciar.
6. Seleccione Actions (Acciones) y, a continuación, elija Reboot node (Reiniciar nodo).

Para reiniciar varios nodos, repita los pasos del 2 al 5 por cada nodo que desee reiniciar. No es necesario esperar a que un nodo termine de reiniciarse para reiniciar otro.

Reinicio de un clúster para Memcached

Al reiniciar un clúster de Memcached, el clúster vacía todos sus datos y reinicia su motor. Durante este proceso, no puede obtener acceso al clúster. Dado que el clúster vacía todos sus datos, cuando este vuelva a estar disponible, empezará con un clúster vacío.

Puede reiniciar un clúster con la consola de ElastiCache, la AWS CLI o la API de ElastiCache. Independientemente de si utiliza la consola de ElastiCache, la AWS CLI o la API de ElastiCache, solo puede reiniciar un solo clúster. Para reiniciar varios clústeres, debe repetir el proceso o la operación.

Uso de Consola de administración de AWS

Puede reiniciar un clúster a través de la consola de ElastiCache.

Para reiniciar un clúster (consola)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En la lista de la esquina superior derecha, elija la región de AWS que le interese.
3. En el panel de navegación, elija el motor que se ejecuta en el clúster que desea reiniciar.

   Se mostrará una lista de los clústeres que ejecutan el motor elegido.

4. Elija el clúster que desea reiniciar activando la casilla situada a la izquierda del nombre del clúster.

   El botón Reiniciar se activa.

   Si elige más de un clúster, el botón Reiniciar se deshabilita.

5. Elija Reboot.

   Aparecerá la pantalla de confirmación de reinicio del clúster.

6. Para reiniciar el clúster, seleccione Reboot (Reiniciar). El estado del clúster cambia a Reiniciar nodos del clúster.

   Para no reiniciar el clúster, seleccione Cancel (Cancelar).

Para reiniciar varios clústeres, repita los pasos del 2 al 5 por cada clúster que desee reiniciar. No es necesario esperar a que un clúster termine de reiniciarse para reiniciar otro.

Para reiniciar un nodo específico, seleccione el nodo y, a continuación, elija Reboot (Reiniciar).

Uso de AWS CLI

Para reiniciar un clúster (AWS CLI), utilice la operación `reboot-cache-cluster` de la CLI.

Para reiniciar nodos específicos en el clúster, use `--cache-node-ids-to-reboot` para enumerar los clústeres específicos que desea reiniciar. El siguiente comando reinicia los nodos 0001, 0002 y 0004 de my-cluster.

Para Linux, macOS o Unix:

```
aws elasticache reboot-cache-cluster \
    --cache-cluster-id my-cluster \
    --cache-node-ids-to-reboot 0001 0002 0004
```

Para Windows:

```
aws elasticache reboot-cache-cluster ^
    --cache-cluster-id my-cluster ^
    --cache-node-ids-to-reboot 0001 0002 0004
```

Para reiniciar todos los nodos del clúster, use el parámetro `--cache-node-ids-to-reboot` y enumere todos los ID del nodo del clúster. Para obtener más información, consulte reboot-cache-cluster.

Uso de la API de ElastiCache

Para reiniciar un clúster mediante la API de ElastiCache, utilice la acción `RebootCacheCluster`.

Para reiniciar nodos específicos en el clúster, use `CacheNodeIdsToReboot` para enumerar los clústeres específicos que desea reiniciar. El siguiente comando reinicia los nodos 0001, 0002 y 0004 de my-cluster.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=RebootCacheCluster
   &CacheClusterId=my-cluster
   &CacheNodeIdsToReboot.member.1=0001
   &CacheNodeIdsToReboot.member.2=0002
   &CacheNodeIdsToReboot.member.3=0004
   &Version=2015-02-02
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &X-Amz-Credential=<credential>
```

Para reiniciar todos los nodos del clúster, use el parámetro `CacheNodeIdsToReboot` y enumere todos los ID del nodo del clúster. Para obtener más información, consulte RebootCacheCluster.

Reemplazo de nodos (Valkey y Redis OSS)

Amazon ElastiCache suele actualizar su flota con parches y actualizaciones que se aplican a las instancias sin problemas. Sin embargo, cada cierto tiempo tenemos que volver a lanzar los nodos de ElastiCache con el fin de aplicar actualizaciones obligatorias de sistemas operativos en el host subyacente. Estas sustituciones son necesarias para aplicar actualizaciones que refuerzan la seguridad, la fiabilidad y el rendimiento operativo.

Puede optar por administrar personalmente estas sustituciones en cualquier momento antes del periodo programado para la sustitución de nodos. Cuando administre personalmente una sustitución, la instancia recibirá la actualización del sistema operativo cuando vuelva a lanzar el nodo y se cancelará la sustitución de nodos programada. Es posible que reciba alertas que indiquen que va a tener lugar la sustitución de nodos. Si ya ha mitigado manualmente la necesidad de mantenimiento, puede hacer caso omiso de estas alertas.

> **Note**
>
> Los nodos de caché de reemplazo generados de forma automática por Amazon ElastiCache pueden tener direcciones IP diferentes. Usted es responsable de revisar la configuración de la aplicación para asegurarse de que los nodos de caché estén asociados con las direcciones IP apropiadas.

En la siguiente lista, se identifican las acciones que puede realizar cuando ElastiCache programa el reemplazo de uno de los nodos de Valkey o Redis OSS. Para agilizar la búsqueda de la información que necesita para su situación, elija una opción del menú siguiente.

- Do nothing: permita que Amazon ElastiCache reemplace el nodo según lo programado.
- Change your maintenance window: cambie el periodo de mantenimiento a un mejor momento.
- Configuraciones de Valkey o Redis OSS (modo de clúster habilitado)
  - Replace the only node in any Valkey or Redis OSS cluster: procedimiento para reemplazar un nodo en un clúster de Valkey o Redis OSS mediante la copia de seguridad y restauración.
  - Replace a replica node in any Valkey or Redis OSS cluster: procedimiento para reemplazar una réplica de lectura en cualquier clúster de Valkey o Redis OSS al aumentar y disminuir el recuento de réplicas sin tiempo de inactividad del clúster.

  - Replace any node in a Valkey or Redis OSS (cluster mode enabled) shard: procedimiento dinámico sin tiempo de inactividad del clúster para reemplazar un nodo en un clúster de Valkey o Redis OSS (modo de clúster habilitado) al escalar y reducir horizontalmente.
- Configuraciones de Valkey o Redis OSS (modo de clúster deshabilitado)
  - Replace the only node in any Valkey or Redis OSS cluster: procedimiento para reemplazar cualquier nodo en un clúster de Valkey o Redis OSS mediante la copia de seguridad y restauración.
  - Replace a replica node in any Valkey or Redis OSS cluster: procedimiento para reemplazar una réplica de lectura en cualquier clúster de Valkey o Redis OSS al aumentar y disminuir el recuento de réplicas sin tiempo de inactividad del clúster.
  - Replace a node in a Valkey or Redis OSS (cluster mode disabled) cluster: procedimiento para reemplazar un nodo en un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) mediante la replicación.
  - Replace a Valkey or Redis OSS (cluster mode disabled) read-replica: procedimiento que permite reemplazar de forma manual una réplica de lectura en un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado).
  - Replace a Valkey or Redis OSS (cluster mode disabled) primary node: procedimiento que permite reemplazar de forma manual el nodo principal en un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado).

Opciones de reemplazo de nodos de Valkey o Redis OSS

- No hacer nada: si no hace nada, ElastiCache reemplaza el nodo según lo programado.

  En las configuraciones que no son de clúster y tienen la conmutación por error automática habilitada, los clústeres de Valkey 7.2 y versiones posteriores y Redis OSS 5.0.6 y versiones posteriores completarán el reemplazo a la vez que el clúster permanece en línea y atiende solicitudes de escritura entrantes. En los clústeres que tienen habilitada la conmutación por error y la versión 4.0.10 de Redis OSS o anteriores, es posible que observe una breve interrupción de escritura que puede llegar a durar unos segundos y que está asociada a las actualizaciones de DNS.

  Si el nodo es miembro de un clúster activado para la conmutación por error automática, ElastiCache para Valkey o Redis OSS proporciona una mayor disponibilidad durante la aplicación

de parches, actualizaciones y otras operaciones de reemplazo de nodos relacionadas con el mantenimiento.

En las configuraciones de clúster de ElastiCache que se han configurado para utilizar clientes del clúster de ElastiCache para Valkey o para Redis OSS, la sustitución se realizará mientras el clúster atiende solicitudes de escritura entrantes.

En las configuraciones que no son de clúster y tienen la conmutación por error automática activada, los clústeres de Valkey 7.2 y versiones posteriores y de Redis OSS 5.0.6 y versiones posteriores completarán el reemplazo a la vez que el clúster permanece en línea y atiende solicitudes de escritura entrantes. En los clústeres que tienen habilitada la conmutación por error y la versión 4.0.10 de Redis OSS o anteriores, es posible que observe una breve interrupción de escritura que puede llegar a durar unos segundos y que está asociada a las actualizaciones de DNS.

Si el nodo es independiente, Amazon ElastiCache primero lanzará un nodo de reemplazo y, a continuación, sincronizará desde el nodo existente. El nodo existente no estará disponible para las solicitudes de servicio durante este periodo. Una vez que se complete la sincronización, el nodo existente se cerrará y el nodo nuevo ocupará su lugar. ElastiCache hará todo lo posible por retener los datos durante esta operación.

- Cambie el periodo de mantenimiento: para eventos de mantenimiento programados, recibirá un correo electrónico o un evento de notificación de ElastiCache. En estos casos, si cambia el periodo de mantenimiento antes de la hora de sustitución programada, su nodo se sustituirá en ese momento a la nueva hora. Para obtener más información, consulte los siguientes temas:
  - Modificación de un clúster de ElastiCache
  - Modificación de un grupo de reproducción

  > **Note**
  >
  > La capacidad de cambiar el periodo de sustitución al mover el periodo de mantenimiento, solo se encuentra disponible cuando la notificación de ElastiCache incluye un periodo de

mantenimiento. Si la notificación no incluye un periodo de mantenimiento, no se puede cambiar el periodo de sustitución.

Por ejemplo, supongamos que es jueves 9 de noviembre a las 15:00 h y el próximo periodo de mantenimiento es el viernes 10 de noviembre a las 17:00 h. A continuación, se exponen tres escenarios con sus resultados:

- Cambia el periodo de mantenimiento a los viernes a las 16:00, después de la fecha y hora actuales y antes del siguiente periodo de mantenimiento programado. El nodo se sustituye el viernes 10 de noviembre a las 16:00.
- Cambia el periodo de mantenimiento al sábado a las 16:00, después de la fecha y hora actuales y después del siguiente periodo de mantenimiento programado. El nodo se sustituye el sábado 11 de noviembre a las 16:00.
- Cambia el periodo de mantenimiento al miércoles a las 16:00 (un día anterior de la misma semana que la fecha y hora actuales). El nodo se sustituye el próximo miércoles 15 de noviembre a las 16:00.

Para obtener instrucciones, consulte Administración del mantenimiento de clústeres de ElastiCache.

- Reemplazar el único nodo en cualquier clúster de Valkey o Redis OSS: si el clúster no tiene réplicas de lectura, puede utilizar el siguiente procedimiento para reemplazar el nodo.

  Para sustituir el único nodo mediante operaciones de copia de seguridad y restauración

  1. Cree una instantánea del clúster del nodo. Para obtener instrucciones, consulte Copias de seguridad manuales.
  2. Cree un nuevo clúster mediante la propagación de los datos a partir de la instantánea. Para obtener instrucciones, consulte Restauración desde una copia de seguridad a una nueva caché.
  3. Elimine el clúster con el nodo programado para su sustitución. Para obtener instrucciones, consulte Eliminación de un clúster en ElastiCache.
  4. En su aplicación, sustituya el punto de enlace del nodo anterior por el punto de enlace del nuevo nodo.

- Reemplazar un nodo de réplica en cualquier clúster de Valkey o Redis OSS: para reemplazar un clúster de réplicas, aumente el recuento de réplicas. Para hacerlo, agregue una réplica y, a continuación, reduzca el número de réplicas eliminando la réplica que desee reemplazar. Este proceso es dinámico y no tiene asociado ningún tiempo de inactividad del clúster.

  > **Note**
  >
  > Si la partición o el grupo de reproducción ya tiene cinco réplicas, invierta los pasos 1 y 2.

  Reemplazo de una réplica en cualquier clúster de Valkey o Redis OSS

  1. Aumente el número de réplicas agregando una réplica al fragmento o grupo de reproducción. Para obtener más información, consulte Aumento del número de réplicas de una partición.
  2. Elimine la réplica que desea sustituir. Para obtener más información, consulte Reducción del número de réplicas de una partición.
  3. Actualice los puntos de enlace de la aplicación.

- Reemplazar cualquier nodo en una partición de Valkey o Redis OSS (modo de clúster habilitado): para reemplazar el nodo de un clúster sin tiempo de inactividad, utilice el cambio de partición en línea. Primero agregue un fragmento mediante el escalado ascendente y, a continuación, elimine el fragmento con el nodo que se reemplazará mediante el escalado descendente.

  Reemplazo de cualquier nodo en un clúster de Valkey o Redis OSS (modo de clúster habilitado)

  1. Escalado ascendente: agregue un fragmento adicional con la misma configuración que el existente con el nodo que se reemplazará. Para obtener más información, consulte Adición de particiones con los cambios de particiones en línea.
  2. Reducción horizontal: elimine la partición con el nodo que se sustituirá. Para obtener más información, consulte Eliminación de particiones con los cambios de particiones en línea.
  3. Actualice los puntos de enlace de la aplicación.

- Reemplazar un nodo en un clúster de Valkey o Redis OSS (modo de clúster deshabilitado): si el clúster es de Valkey o Redis OSS (modo de clúster deshabilitado) sin réplicas de lectura, utilice el siguiente procedimiento para reemplazar el nodo.

  Para sustituir el nodo mediante la reproducción (solo modo de clúster desactivado), realice el siguiente procedimiento:

  1. Agregue la reproducción al clúster con el nodo programado para reemplazo como nodo primario. No habilite Multi-AZ en este clúster. Para obtener instrucciones, consulte Adición de replicación a un clúster de Valkey o Redis OSS sin particiones.
  2. Agregue una réplica de lectura al clúster. Para obtener instrucciones, consulte Adición de nodos a un clúster de ElastiCache (consola).
  3. Promocione la réplica de lectura recién creada a nodo principal. Para obtener instrucciones, consulte Promoción de una réplica de lectura a principal para grupos de replicación de Valkey o Redis OSS (modo de clúster deshabilitado).
  4. Elimine el nodo programado para su sustitución. Para obtener instrucciones, consulte Eliminación de nodos de un clúster de ElastiCache.
  5. En su aplicación, sustituya el punto de enlace del nodo anterior por el punto de enlace del nuevo nodo.

- Reemplazar una réplica de lectura de Valkey o Redis OSS (clúster modo deshabilitado): si el nodo es una réplica de lectura, reemplácelo.

  Si su clúster tiene solo un nodo de réplica y están habilitadas Multi-AZ, debe deshabilitarlas para poder eliminar la réplica. Para obtener instrucciones, consulte Modificación de un grupo de reproducción.

  Reemplazo de una réplica de lectura de Valkey o Redis OSS (modo de clúster deshabilitado)

  1. Elimine la réplica programada para la sustitución. Para obtener instrucciones, consulte lo siguiente:
     - Reducción del número de réplicas de una partición
     - Eliminación de nodos de un clúster de ElastiCache

2. Agregue una nueva réplica para reemplazar la programada para el reemplazo. Si utiliza el mismo nombre que el de la réplica que acaba de eliminar, puede pasar al paso 3. Para obtener instrucciones, consulte lo siguiente:
   - Aumento del número de réplicas de una partición
   - Adición de una réplica de lectura para Valkey o Redis OSS (modo de clúster deshabilitado)
3. En su aplicación, sustituya el punto de enlace de la réplica anterior por el punto de enlace de la nueva réplica.
4. Si deshabilitó las Multi-AZ al inicio, vuelva a habilitarlas ahora. Para obtener instrucciones, consulte Habilitación de Multi-AZ .

- Reemplazar un nodo principal de Valkey o Redis OSS (modo de clúster deshabilitado): si es el nodo principal, primero promueva una réplica de lectura a principal. A continuación, elimine la réplica que era el nodo primario.

  Si su clúster tiene solo una réplica y están habilitadas Multi-AZ, debe deshabilitarlas para poder eliminar la réplica en el paso 2. Para obtener instrucciones, consulte Modificación de un grupo de reproducción.

  Reemplazo de un nodo principal de Valkey o Redis OSS (modo de clúster deshabilitado)

  1. Promocione una réplica de lectura a primaria. Para obtener instrucciones, consulte Promoción de una réplica de lectura a principal para grupos de replicación de Valkey o Redis OSS (modo de clúster deshabilitado).
  2. Elimine el nodo cuya sustitución está programada (antiguo nodo primario). Para obtener instrucciones, consulte Eliminación de nodos de un clúster de ElastiCache.
  3. Agregue una nueva réplica para reemplazar la programada para el reemplazo. Si utiliza el mismo nombre que el del nodo que acaba de eliminar, puede omitir el cambio de puntos de enlace en la aplicación.

     Para obtener instrucciones, consulte Adición de una réplica de lectura para Valkey o Redis OSS (modo de clúster deshabilitado).
  4. En su aplicación, sustituya el punto de enlace del nodo anterior por el punto de enlace del nuevo nodo.

5. Si deshabilitó las Multi-AZ al inicio, vuelva a habilitarlas ahora. Para obtener instrucciones, consulte Habilitación de Multi-AZ .

Reemplazo de nodos (Memcached)

Amazon ElastiCache for Memcached suele actualizar su flota con parches y actualizaciones que se aplican a las instancias sin problemas. Sin embargo, cada cierto tiempo tenemos que volver a lanzar los nodos de ElastiCache for Memcached con el fin de aplicar actualizaciones obligatorias de sistemas operativos en el anfitrión subyacente. Estas sustituciones son necesarias para aplicar actualizaciones que refuerzan la seguridad, la fiabilidad y el rendimiento operativo.

Puede optar por administrar personalmente estas sustituciones en cualquier momento antes del periodo programado para la sustitución de nodos. Cuando administre personalmente una sustitución, la instancia recibirá la actualización del sistema operativo cuando vuelva a lanzar el nodo y se cancelará la sustitución de nodos programada. Es posible que reciba alertas que indiquen que va a tener lugar la sustitución de nodos. Si ya ha mitigado manualmente la necesidad de mantenimiento, puede hacer caso omiso de estas alertas.

> **Note**
>
> Los nodos de caché de reemplazo generados de forma automática por Amazon ElastiCache pueden tener direcciones IP diferentes. Usted es responsable de revisar la configuración de la aplicación para asegurarse de que los nodos de caché estén asociados con las direcciones IP apropiadas.

En la siguiente lista se identifican las acciones que puede realizar cuando ElastiCache programa el reemplazo de uno de los nodos de Memcached.

- No hacer nada: si no hace nada, ElastiCache reemplaza el nodo según lo programado. Cuando ElastiCache reemplaza de forma automática el nodo por uno nuevo, este se encontrará vacío inicialmente.
- Cambie el periodo de mantenimiento: para eventos de mantenimiento programados, recibirá un correo electrónico o un evento de notificación de ElastiCache. En este caso, si cambia el periodo de mantenimiento antes de la hora de sustitución programada, su nodo se sustituirá en ese momento a la nueva hora. Para obtener más información, consulte Modificación de un clúster de ElastiCache.

> **Note**
>
> La capacidad de cambiar el periodo de sustitución al mover el periodo de mantenimiento, solo se encuentra disponible cuando la notificación de ElastiCache incluye un periodo de mantenimiento. Si la notificación no incluye un periodo de mantenimiento, no se puede cambiar el periodo de sustitución.

Por ejemplo, supongamos que es jueves 9 de noviembre a las 15:00 h y el próximo periodo de mantenimiento es el viernes 10 de noviembre a las 17:00 h. A continuación, se exponen tres escenarios con sus resultados:

- Cambia el periodo de mantenimiento a los viernes a las 16:00, después de la fecha y hora actuales y antes del siguiente periodo de mantenimiento programado. El nodo se sustituye el viernes 10 de noviembre a las 16:00.
- Cambia el periodo de mantenimiento al sábado a las 16:00, después de la fecha y hora actuales y después del siguiente periodo de mantenimiento programado. El nodo se sustituye el sábado 11 de noviembre a las 16:00.
- Cambia el periodo de mantenimiento al miércoles a las 16:00 (un día anterior de la misma semana que la fecha y hora actuales). El nodo se sustituye el próximo miércoles 15 de noviembre a las 16:00.

Para obtener instrucciones, consulte Administración del mantenimiento de clústeres de ElastiCache.

- Reemplazar el nodo de forma manual: si necesita reemplazar el nodo antes del siguiente periodo de mantenimiento, hágalo de forma manual.

  Si sustituye manualmente el nodo, las claves se redistribuyen. Esa redistribución provoca errores en la caché.

  Para sustituir manualmente un nodo de Memcached

  1. Elimine el nodo programado para su sustitución. Para obtener instrucciones, consulte Eliminación de nodos de un clúster de ElastiCache.
  2. Agregue un nuevo nodo al clúster. Para obtener instrucciones, consulte Adición de nodos a un clúster de ElastiCache.

3. Si no va a utilizar la detección automática en este clúster, consulte la aplicación y sustituya todas las instancias del punto de conexión del nodo antiguo por el punto de conexión del nuevo nodo.

Nodos reservados

Reservar uno o varios nodos de ElastiCache puede ser una forma de reducir los costos. Los nodos reservados se cobran por adelantado en función de una tarifa que depende del tipo de nodo y la duración de la reserva: uno o tres años.

Para ver si los nodos reservados suponen un ahorro de costos para sus casos de uso, determine primero el tamaño del nodo y la cantidad de nodos que necesita. A continuación, calcule la utilización del nodo y compare el costo total que supondría utilizar nodos en diferido en relación con los nodos reservados. Puede mezclar y emparejar nodos reservados con nodos en diferido en sus clústeres. Para obtener información sobre los precios, consulte Precios de Amazon ElastiCache.

Temas

- Administración de costos con los nodos reservados
- Ofertas de nodos reservados estándar
- Tamaño de los nodos reservados con flexibilidad
- Eliminación de un nodo reservado
- Ofertas de nodos reservados heredados
- Obtención de información sobre las ofertas de nodos reservados
- Compra de un nodo reservado
- Obtención de información sobre los nodos reservados

Administración de costos con los nodos reservados

Reservar uno o varios nodos puede ser una forma de reducir costos. Los nodos reservados se cobran por adelantado en función de una tarifa que depende del tipo de nodo y la duración de la reserva: uno o tres años. Dicho cargo es mucho menor que el cargo por uso por hora que se aplica con los nodos bajo demanda.

Para ver si los nodos reservados suponen un ahorro de costos para sus casos de uso, determine primero el tamaño del nodo y la cantidad de nodos que necesita. A continuación, calcule el uso del nodo y compare el costo total que supondría utilizar nodos bajo demanda en relación con los nodos

reservados. Puede mezclar y emparejar nodos reservados con nodos en diferido en sus clústeres. Para obtener información sobre los precios, consulte Precios de Amazon ElastiCache.

La región de AWS, el tipo de nodo y la longitud del término se deben elegir en el momento de la compra y no se pueden cambiar más adelante.

Puede utilizar la Consola de administración de AWS, la AWS CLI o la API de ElastiCache para mostrar y adquirir ofertas de nodos reservados disponibles.

Para obtener más información sobre los nodos reservados, consulte Nodos reservados de Amazon ElastiCache.

Ofertas de nodos reservados estándar

Cuando compra una instancia de nodo reservado (RI) en Amazon ElastiCache, puede adquirir un compromiso para obtener una tarifa con descuento en un tipo de instancias de nodo específico y una región de AWS por la duración de la instancia de nodo reservada. Para utilizar una instancia reservada de nodo de Amazon ElastiCache, debe crear una instancia de nodo de ElastiCache nueva, al igual que lo haría para una instancia en diferido.

Si las especificaciones de la instancia de nodo reservado nueva coinciden con una instancia reservada de nodo existente de su cuenta, se facturará a la tarifa con descuento que se ofrece para la instancia de nodo reservada. De lo contrario, la instancia de nodo se factura con una tarifa en diferido. Estas RI estándar están disponibles a partir de las familias de instancias R5 y M5 en adelante.

> **Note**
>
> Los tipos de ofertas que se analizan a continuación están disponibles en plazos de un año y de tres años.

Tipos de ofertas

Una RI sin pago inicial proporciona acceso a una instancia reservada de ElastiCache sin que haya que hacer un pago inicial. La instancia reservada de ElastiCache sin pago inicial se factura con una tarifa por hora con descuento por cada hora dentro del término, independientemente de si se utiliza o no.

Una RI con pago inicial parcial requiere que parte de la instancia reservada de ElastiCache se pague por adelantado. Las horas restantes del término se cobran a una tarifa por hora con descuento,

independientemente de la utilización que haga. Esta opción sustituye a la opción heredada utilización intensa, que se explica en la sección siguiente.

La RI de Pago total anticipado requiere que el pago completo se realice al comenzar el término de RI. No se aplicarán otros costos durante el resto del término, independientemente del número de horas utilizadas.

Tamaño de los nodos reservados con flexibilidad

Todos los nodos reservados tienen un tamaño flexible. Al comprar un nodo reservado, una de las cosas que especifica es el tipo de nodo, por ejemplo, cache.r6g.xlarge. Para obtener más información sobre los tipos de nodos, consulte Precios de Amazon ElastiCache.

Si tiene un nodo y debe escalarlo para aumentar su capacidad, el nodo reservado se aplica automáticamente al nodo escalado. Es decir, los nodos reservados se aplican automáticamente al uso de cualquier tamaño en la misma familia de nodos. Los nodos reservados de tamaño flexible están disponibles para los nodos de la misma región de AWS. Los nodos reservados con flexibilidad de tamaño solo se pueden reducir horizontalmente en sus familias de nodos. Por ejemplo, un nodo reservado para cache.r6g.xlarge puede aplicarse a cache.r6g.2xlarge, pero no a cache.r6gd.large, porque cache.r6g y cache.r6gd son familias de nodos diferentes.

La flexibilidad de tamaño significa que puede moverse libremente entre configuraciones dentro de la misma familia de nodos. Por ejemplo, puede pasar de un nodo reservado r6g.xlarge (8 unidades normalizadas) a dos nodos reservados r6g.large (8 unidades normalizadas) (2 × 4 = 8 unidades normalizadas) en la misma región de AWS sin costo adicional.

Actualización de nodos de Redis OSS a Valkey

Con el lanzamiento de Valkey en ElastiCache, ahora puede aplicar su descuento en nodos reservados de Redis OSS al motor de caché de Valkey. Puede actualizar de Redis OSS a Valkey sin dejar de beneficiarse de los contratos y reservas existentes. Además de poder aprovechar sus ventajas dentro de la familia de nodos de caché y del motor, puede obtener también un mayor valor incremental. Valkey tiene un descuento del 20 % en comparación con Redis OSS y, con la flexibilidad de los nodos reservados, puede utilizar sus nodos reservados de Redis OSS para cubrir un 20 % más de nodos de Valkey en ejecución.

Para calcular la tarifa con descuento, cada combinación de nodo y motor de ElastiCache tiene un factor de normalización que se mide en unidades. Las unidades de nodos reservados se pueden aplicar a cualquier nodo en ejecución de la familia de instancias del nodo reservado para un motor determinado. Los nodos reservados de Redis OSS también se pueden aplicar a todos los motores

para cubrir los nodos de Valkey en ejecución. Como Valkey tiene un descuento en comparación con Redis OSS y Memcached, sus unidades para un tipo de instancia determinado son más bajas, lo que permite que un nodo reservado de Redis OSS cubra más nodos de Valkey.

Como ejemplo, supongamos que ha comprado un nodo reservado para una cache.r7g.4xlarge para el motor de Redis OSS (32 unidades) y está ejecutando un nodo cache.r7g.4xlarge de Redis OSS (32 unidades). Si actualiza el nodo a Valkey, el factor de normalización del nodo en ejecución se reduce a 25,6 unidades y el nodo reservado existente le proporciona 6,4 unidades adicionales para utilizarlas con cualquier otro nodo de Valkey o Redis OSS en ejecución de la familia cache.r7g en la región. Puede usar esto para cubrir el 25 % de otro nodo de Valkey cache.r7g.4xlarge de la cuenta (25,6 unidades) o el 100 % de un nodo de Valkey cache.r7g.xlarge (6,4 unidades).

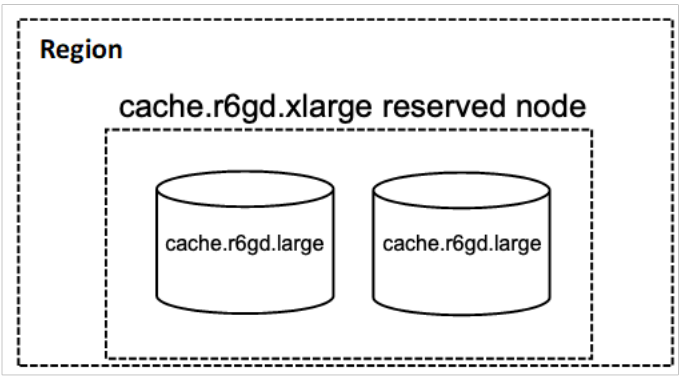
Comparación del uso con unidades normalizadas

Puede comparar el uso de diferentes tamaños de nodos reservados utilizando unidades normalizadas. Por ejemplo, una hora de uso en dos nodos cache.r6g.4xlarge equivale a 16 horas de uso en una sola cache.r6g.large. En la tabla siguiente se muestra el número de unidades normalizadas por cada tamaño de nodo:

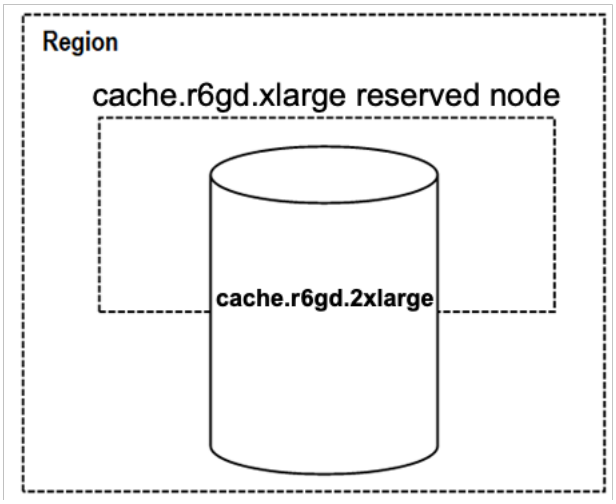
Tamaño del nodo	Unidades normalizadas con Redis OSS o Memcached	Unidades normalizadas con Valkey
micro	0,5	0,4
small	1	0,8
medium	2	1,6
large	4	3,2
xlarge	8	6,4
2xlarge	16	12,8
4xlarge	32	25,6
6xlarge	48	38,4
8xlarge	64	51,2

Tamaño del nodo	Unidades normalizadas con Redis OSS o Memcached	Unidades normalizadas con Valkey
10xlarge	80	64
12xlarge	96	76,8
16xlarge	128	102,4
24xlarge	192	153,6

Por ejemplo, adquiere un nodo reservado cache.r6gd.xlarge y tiene dos nodos reservados cache.r6gd.large en ejecución en su cuenta en la misma región de AWS. En este caso, el beneficio de facturación se aplica en su totalidad a los dos nodos.



De forma alternativa, si tiene una instancia cache.r6gd.2xlarge en ejecución en su cuenta en la misma región de AWS, el beneficio en la facturación se aplica al 50 % del uso del nodo reservado.



Eliminación de un nodo reservado

Los términos de un nodo reservado implican un compromiso de un año o de tres años. No se puede cancelar un nodo reservado. Sin embargo, puede eliminar un nodo que tenga un descuento de nodo reservado. El proceso para eliminar un nodo con un descuento de nodo reservado es el mismo que para cualquier otro nodo.

Si elimina un nodo con un descuento de nodo reservado, puede lanzar otro nodo con especificaciones compatibles. En este caso, sigue disfrutando de la tarifa de descuento mientras dure la reserva (de uno o tres años).

Ofertas de nodos reservados heredados

Existen tres niveles de reservas de nodo heredado: uso intenso, uso medio y uso bajo. Los nodos pueden reservarse en cualquier nivel de uso durante uno o tres años. El tipo de nodo, el nivel de uso y el término de la reserva afectarán a los costos totales. Verifique el ahorro que los nodos reservados pueden proporcionar a la empresa al comparar varios modelos antes de adquirir los nodos reservados.

Los nodos que se compran con un término o nivel de utilización determinado no pueden cambiarse a un término o nivel de utilización distinto.

Niveles de utilización

Los nodos reservados de utilización intensa permiten cargas de trabajo que tienen una base constante de capacidad, así como la ejecución de cargas de trabajo estables. Los nodos reservados de utilización intensa requieren un elevado nivel de compromiso por anticipado; sin embargo, si planea mantener el nodo en ejecución más del 79 por ciento del término de la reserva del nodo, podrá conseguir un mayor ahorro (hasta un 70 por ciento en comparación con el precio de los nodos en diferido). Con los nodos reservados de uso intenso, deberá realizar un pago por única vez. A continuación, seguirá una tarifa por hora más baja mientras dure el término, independientemente de si el nodo se encuentre en funcionamiento.

Los nodos reservados de uso medio son la mejor opción si tiene previsto aprovechar los nodos reservados durante una buena parte del tiempo, pero desea disponer de una tarifa de pago por única vez menor o de la flexibilidad para poder dejar de pagar por el nodo al apagarlo. Los nodos reservados de utilización media son una opción más rentable cuando tiene planeado mantener en ejecución los nodos reservados más del 40 por ciento del término que dura la reserva. Esta opción puede ahorrar hasta un 64 por ciento con respecto al precio en diferido. Con los nodos reservados

de utilización media, el pago por única vez será ligeramente superior que con los nodos reservados de utilización baja y disfrutará de unas tarifas de uso por hora más reducidas al ejecutar un nodo.

Los nodos reservados de utilización baja son la opción perfecta para cargas de trabajo periódicas que solo se ejecutan durante un par de horas al día o durante unos días a la semana. Con los nodos reservados de utilización baja, deberá realizar un pago por única vez seguido del pago de una tarifa de utilización por horas con descuento cuando el nodo se ejecute. Puede comenzar a ahorrar cuando el nodo se ejecute durante más del 17 % del término reservado del nodo. Puede ahorrar hasta un 56 % de las tarifas bajo demanda durante todo el término del nodo reservado.

Ofertas de nodos reservados heredados

Ofertas	Costo inicial	Tarifa de uso	Beneficio
Utilización intensa	Máxima	Tasa por hora más baja. Se aplica durante todo el término independi entemente de si utiliza el nodo reservado o no.	Menor costo total si planea ejecutar los nodos reservados más del 79 por ciento del término de tres años.
Utilización media	Medio	Tasa de utilización por hora aplicada por cada hora de ejecución del nodo. No se aplica ningún cargo por hora cuando el nodo no está en ejecución.	Adecuado para cargas de trabajo elásticas o para cuando se espera una utilización moderada, durante más del 40 por ciento del término de tres años.
Utilización baja	Mínima	Tasa de uso por hora aplicada por cada hora de ejecución del nodo. No se aplica ningún cargo por hora cuando	El costo total más elevado si planea ejecutar todo el tiempo. Sin embargo, ofrece el mínimo costo total si prevé

Ofertas	Costo inicial	Tarifa de uso	Beneficio
		el nodo no está en ejecución. Tasa por hora más alta de todos los tipos de ofertas; sin embargo, las tasas solo se aplican cuando el nodo reservado está en ejecución.	que utilizará el nodo reservado con poca frecuencia, más del 15 % de un término de tres años.
Utilización en diferido (sin nodos reservados)	Ninguno	Tasa por hora más alta. Se aplica cuando el nodo está en ejecución.	Costo por hora más alto.

Para obtener más información, consulte Precios de Amazon ElastiCache.

Obtención de información sobre las ofertas de nodos reservados

Antes de comprar nodos reservados, puede obtener información acerca de las ofertas de nodos reservados.

Los siguientes ejemplos muestran cómo obtener datos de precios e información sobre las ofertas disponibles de nodos reservados mediante la Consola de administración de AWS, la AWS CLI y la API de ElastiCache.

Temas

- Obtención de información sobre las ofertas de nodos reservados (consola)
- Obtención de información sobre las ofertas de nodos reservados (AWS CLI)
- Obtención de información sobre las ofertas de nodos reservados (API de ElastiCache)

Obtención de información sobre las ofertas de nodos reservados (consola)

Para obtener datos de precios e información sobre las ofertas disponibles de clústeres reservados mediante la Consola de administración de AWS, utilice el siguiente procedimiento.

Para obtener información acerca de las ofertas de nodos reservados disponibles

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, seleccione Reserved Nodes (Nodos reservados).
3. Elija Purchase Reserved Node (Comprar nodo reservado).
4. En Motor, elija Valkey, Memcached o Redis OSS.
5. Para determinar las ofertas disponibles, seleccione las siguientes opciones:

   - Node Type
   - Term (Término
   - Offering Type(Tipo de oferta

   Después de realizar estas selecciones, el costo por nodo y el costo total de las selecciones se mostrarán en Reservation details (Detalles de reserva).
6. Elija Cancel (Cancelar) para evitar comprar estos nodos e incurrir en gastos.

Obtención de información sobre las ofertas de nodos reservados (AWS CLI)

Para obtener más información acerca de precios y otra información acerca de las ofertas de nodos reservados disponibles para Valkey o Redis OSS, escriba el comando siguiente en el símbolo del sistema:

```
aws elasticache describe-reserved-cache-nodes-offerings
```

Esta operación produce un resultado similar al siguiente (formato JSON):

```
{
          "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
          "CacheNodeType": "cache.xxx.large",
          "Duration": 94608000,
          "FixedPrice": XXXX.X,
          "UsagePrice": X.X,
          "ProductDescription": "redis",
          "OfferingType": "All Upfront",
          "RecurringCharges": [
              {
                  "RecurringChargeAmount": X.X,
                  "RecurringChargeFrequency": "Hourly"
              }
          ]
 },
 {
          "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
          "CacheNodeType": "cache.xxx.xlarge",
          "Duration": 94608000,
          "FixedPrice": XXXX.X,
          "UsagePrice": X.X,
          "ProductDescription": "redis",
          "OfferingType": "Partial Upfront",
          "RecurringCharges": [
              {
                  "RecurringChargeAmount": X.XXX,
                  "RecurringChargeFrequency": "Hourly"
              }
          ]
 },
 {
          "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
          "CacheNodeType": "cache.xxx.large",
```

```
            "Duration": 31536000,
            "FixedPrice": X.X,
            "UsagePrice": X.X,
            "ProductDescription": "redis",
            "OfferingType": "No Upfront",
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": X.XXX,
                    "RecurringChargeFrequency": "Hourly"
                }
            ]
}
```

Para obtener más información acerca de precios y otra información acerca de las ofertas de nodos reservados disponibles, escriba el comando siguiente en el símbolo del sistema:

```
 {
            "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
            "CacheNodeType": "cache.xxx.large",
            "Duration": 94608000,
            "FixedPrice": XXXX.X,
            "UsagePrice": X.X,
            "ProductDescription": "memcached",
            "OfferingType": "All Upfront",
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": X.X,
                    "RecurringChargeFrequency": "Hourly"
                }
            ]
    },
 {
            "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
            "CacheNodeType": "cache.xxx.xlarge",
            "Duration": 94608000,
            "FixedPrice": XXXX.X,
            "UsagePrice": X.X,
            "ProductDescription": "memcached",
            "OfferingType": "Partial Upfront",
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": X.XXXX,
                    "RecurringChargeFrequency": "Hourly"
```

```
                }
            ]
        },
        {
            "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
            "CacheNodeType": "cache.xx.12xlarge",
            "Duration": 31536000,
            "FixedPrice": X.X,
            "UsagePrice": X.X,
            "ProductDescription": "memcached",
            "OfferingType": "No Upfront",
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": X.XXXX,
                    "RecurringChargeFrequency": "Hourly"
                }
            ]
}
```

Para obtener más información, consulte describe-reserved-cache-nodes-offerings en la referencia de la AWS CLI.

Obtención de información sobre las ofertas de nodos reservados (API de ElastiCache)

Para obtener datos acerca de precios e información acerca de las ofertas de nodos reservados disponibles, realice una llamada a la acción `DescribeReservedCacheNodesOfferings`.

Example

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=DescribeReservedCacheNodesOfferings
    &Version=2014-12-01
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &X-Amz-Algorithm
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Para obtener más información, consulte DescribeReservedCacheNodesOfferings en la referencia de la API de ElastiCache.

Compra de un nodo reservado

Los siguientes ejemplos muestran cómo comprar una oferta de nodo reservado mediante la Consola de administración de AWS, la AWS CLI y la API de ElastiCache.

> **Important**
>
> Los ejemplos que se muestran a continuación en esta sección incurrirán en gastos en la cuenta de AWS que no podrá revertir.

Temas

- Compra de un nodo reservado (Consola)
- Compra de un nodo reservado (AWS CLI)
- Compra de un nodo reservado (API de ElastiCache)

Compra de un nodo reservado (Consola)

En este ejemplo, se muestra la compra de una oferta de nodo reservado específica, 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f, con un ID de nodo reservado de myreservationID.

En el procedimiento siguiente, se utiliza la Consola de administración de AWS para comprar la oferta de nodo reservado por identificador de oferta.

Para comprar nodos reservados

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija el enlace Reserved Nodes (Nodos reservados).
3. Elija el botón Purchase reserved nodes (Comprar nodos reservados).
4. En Motor, elija Valkey, Memcached o Redis OSS.
5. Para determinar las ofertas disponibles, seleccione las siguientes opciones:
    - Node Type
    - Term (Término
    - Offering Type(Tipo de oferta

- Un Reserved node ID (ID de nodo reservado) opcional

Después de realizar estas selecciones, el costo por nodo y el costo total de las selecciones se mostrarán en Reservation details (Detalles de reserva).

6. Seleccione Comprar.

Compra de un nodo reservado (AWS CLI)

En el siguiente ejemplo, se muestra la compra de una oferta de clúster reservado específica, 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f, con un ID de nodo reservado de myreservationID.

Escriba el comando siguiente en el símbolo del sistema:

Para Linux, macOS o Unix:

```
aws elasticache purchase-reserved-cache-nodes-offering \
    --reserved-cache-nodes-offering-id 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f \
    --reserved-cache-node-id myreservationID
```

Para Windows:

```
aws elasticache purchase-reserved-cache-nodes-offering ^
    --reserved-cache-nodes-offering-id 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f ^
    --reserved-cache-node-id myreservationID
```

El comando devuelve un resultado similar al siguiente:

```
RESERVATION  ReservationId      Class           Start Time                Duration
 Fixed Price  Usage Price  Count  State           Description      Offering Type
RESERVATION  myreservationid    cache.xx.small  2013-12-19T00:30:23.247Z  1y
 XXX.XX USD   X.XXX USD    1      payment-pending  memcached        Medium Utilization
```

Para obtener más información, consulte purchase-reserved-cache-nodes-offering en la referencia de la AWS CLI.

Compra de un nodo reservado (API de ElastiCache)

En el siguiente ejemplo, se muestra la compra de una oferta de nodo reservado específica, 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f, con un ID de clúster reservado de myreservationID.

Realice una llamada a la operación `PurchaseReservedCacheNodesOffering` con los parámetros siguientes:

- `ReservedCacheNodesOfferingId` = `649fd0c8-cf6d-47a0-bfa6-060f8e75e95f`
- `ReservedCacheNodeID` = `myreservationID`
- `CacheNodeCount` = `1`

Example

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=PurchaseReservedCacheNodesOffering
    &ReservedCacheNodesOfferingId=649fd0c8-cf6d-47a0-bfa6-060f8e75e95f
    &ReservedCacheNodeID=myreservationID
    &CacheNodeCount=1
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20141201T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Para obtener más información, consulte PurchaseReservedCacheNodesOffering en la referencia de la API de ElastiCache.

Obtención de información sobre los nodos reservados

Puede obtener información sobre los nodos reservados que ha comprado con la Consola de administración de AWS, la AWS CLI y la API de ElastiCache.

Temas

- Obtención de información sobre los nodos reservados (consola)
- Obtención de información sobre los nodos reservados (AWS CLI)
- Obtención de información sobre los nodos reservados (API de ElastiCache)

Obtención de información sobre los nodos reservados (consola)

En el procedimiento siguiente, se describe cómo utilizar la Consola de administración de AWS para obtener información acerca de los nodos reservados que ha comprado.

Para obtener información acerca de los nodos reservados comprados

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija el enlace Reserved nodes (Nodos reservados).

   Los nodos reservados para la cuenta aparecen en la lista de nodos reservados. Puede elegir cualquiera de los nodos reservados de la lista para ver información detallada acerca del nodo reservado en el panel de detalles situado en la parte inferior de la consola.

Obtención de información sobre los nodos reservados (AWS CLI)

Para obtener información sobre los nodos reservados de la cuenta de AWS, escriba el siguiente comando en el símbolo del sistema:

```
aws elasticache describe-reserved-cache-nodes
```

Esta operación produce un resultado similar al siguiente (formato JSON):

```
{
    "ReservedCacheNodeId": "myreservationid",
    "ReservedCacheNodesOfferingId": "649fd0c8-cf6d-47a0-bfa6-060f8e75e95f",
    "CacheNodeType": "cache.xx.small",
```

```
    "DataTiering": "disabled",
    "Duration": "31536000",
    "ProductDescription": "memcached",
    "OfferingType": "Medium Utilization",
    "MaxRecords": 0
}
```

Para obtener más información, consulte describe-reserved-cache-nodes en la referencia de la AWS CLI.

Obtención de información sobre los nodos reservados (API de ElastiCache)

Para obtener información sobre los nodos reservados de la cuenta de AWS, realice una llamada a la operación `DescribeReservedCacheNodes`.

Example

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=DescribeReservedCacheNodes
    &Version=2014-12-01
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20141201T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Para obtener más información, consulte DescribeReservedCacheNodes en la referencia de la API de ElastiCache.

Migración de nodos de generación anterior

Los nodos de generación anterior son tipos de nodos que se eliminan gradualmente. Si no tiene clústeres existentes que utilicen un tipo de nodo de generación anterior, ElastiCache no admite la creación de clústeres nuevos con ese tipo de nodo.

Debido a la cantidad limitada de tipos de nodos de generación anterior, no podemos garantizar un reemplazo exitoso cuando un nodo pasa a estar en mal estado en los clústeres. En tal situación, la disponibilidad del clúster puede verse afectada de forma negativa.

Recomendamos migrar los clústeres a un tipo de nodo nuevo para mejorar la disponibilidad y el rendimiento. Para obtener un tipo de nodo recomendado que permita migrar, consulte Rutas de actualización. Para obtener una lista completa de los tipos de nodos admitidos y los tipos de nodos de generación anterior en ElastiCache, consulte Tipos de nodos compatibles.

Migración de nodos en un clúster de Valkey o Redis OSS

El siguiente procedimiento describe cómo migrar el tipo de nodo del clúster de Valkey o Redis OSS con la consola de ElastiCache. Durante este proceso, el clúster de Valkey o Redis OSS seguirá atendiendo solicitudes con un tiempo de inactividad mínimo. Según la configuración del clúster, puede ver los siguientes tiempos de inactividad. Las siguientes son estimaciones y pueden diferir en función de sus configuraciones específicas:

- El modo de clúster deshabilitado (nodo único) puede ver aproximadamente 60 segundos, principalmente debido a la propagación de DNS.
- El modo de clúster deshabilitado (con nodo de réplica) puede experimentar aproximadamente 1 segundo para los clústeres que ejecutan Valkey 7.2 y versiones posteriores o Redis OSS 5.0.6 y versiones posteriores. Todas las versiones inferiores pueden experimentar aproximadamente 10 segundos.
- El modo de clúster habilitado puede ver aproximadamente 1 segundo.

Para modificar un tipo de nodo de clúster de Valkey o Redis OSS mediante la consola:

1. Inicie sesión en la consola y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Clústeres de Valkey o Clústeres de Redis OSS.
3. En la lista de clústeres, seleccione el clúster que desea migrar.
4. Elija Actions (Acciones) y después Modify (Modificar).
5. Elija el nuevo tipo de nodo de la lista de tipos de nodos.
6. Si desea realizar el proceso de migración de forma inmediata, elija Apply immediately (Aplicar de inmediato). Si no selecciona Apply immediately (Aplicar de inmediato), el proceso de migración se realizará durante el siguiente periodo de mantenimiento del clúster.
7. Elija Modify (Modificar). Si eligió Apply immediately (Aplicar inmediatamente) en el paso anterior, el estado del clúster cambiará a modifying (en modificación). Cuando el estado cambie a available (disponible), la modificación se habrá completado y podrá empezar a utilizar el nuevo clúster.

Para modificar un tipo de nodo de clúster de Valkey o Redis OSS mediante la AWS CLI:

Utilice la API modify-replication-group como se muestra a continuación:

Para Linux, macOS o Unix:

```
aws elasticache modify-replication-group /
 --replication-group-id my-replication-group /
 --cache-node-type new-node-type /
 --apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group ^
 --replication-group-id my-replication-group ^
 --cache-node-type new-node-type ^
 --apply-immediately
```

En esta situación, el valor de *new-node-type* es el tipo de nodo al que se migra. Al pasar el parámetro `--apply-immediately`, la actualización se aplicará de inmediato cuando el grupo de reproducción pase del estado de modifying (modificando) a modificación (disponible). Si no selecciona Apply immediately (Aplicar de inmediato), el proceso de migración se realizará durante el siguiente periodo de mantenimiento del clúster.

> **Note**
>
> Si no puede modificar el clúster con un error `InvalidCacheClusterState`, primero debe eliminar un nodo con error de restauración.

Reparación o eliminación de los nodos con error en la restauración

El siguiente procedimiento describe cómo reparar o eliminar los nodos con error de restauración en el clúster de Valkey o Redis OSS. Para obtener más información sobre cómo entran los nodos de ElastiCache en un estado de restauración fallida, consulte Visualización del estado de los nodos en ElastiCache. Es recomendable eliminar primero los nodos que se encuentren en un estado de restauración fallida; luego, migrar los nodos restantes de la generación anterior del clúster de

ElastiCache a un tipo de nodo de nueva generación y, por último, volver a añadir el número de nodos necesario.

Eliminar el nodo con error de restauración (consola):

1. Inicie sesión en la consola y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Clústeres de Valkey o Clústeres de Redis OSS.
3. En la lista de clústeres, elija el clúster del que desea eliminar un nodo.
4. En la lista de particiones, elija la partición de la que desea eliminar un nodo. Omita este paso si el modo clúster se encuentra deshabilitado para el clúster.
5. En la lista de nodos, elija el nodo con un estado `restore-failed`.
6. Elija Actions (Acciones) y, a continuación, elija Delete node (Eliminar nodo).

Cuando elimine los nodos con errores de restauración del clúster de ElastiCache, ya puede migrar a un tipo de generación más reciente. Para obtener más información, consulte lo descrito anteriormente sobre Migración de nodos en un clúster de Valkey o Redis OSS.

Para añadir nodos posteriores al clúster de ElastiCache, consulte Adición de nodos a un clúster de ElastiCache.

Migración de nodos en un clúster de Memcached

Para migrar ElastiCache para Memcached a un tipo de nodo distinto, debe crear un clúster nuevo que siempre comience vacío así la aplicación puede rellenarlo.

Para migrar el tipo de nodo de clúster de ElastiCache para Memcached mediante la consola de ElastiCache:

- Cree un nuevo clúster con el nuevo tipo de nodo. Para obtener más información, consulte Creación de un clúster de Memcached (consola).
- En la aplicación, actualice los puntos de enlace a los puntos de enlace del nuevo clúster. Para obtener más información, consulte Búsqueda de puntos de conexión de un clúster (consola) (Memcached)
- Elimine el clúster anterior. Para obtener más información, consulte Eliminación de un clúster en ElastiCache

Administración de clústeres en ElastiCache

Un clúster es una colección de uno o varios nodos de caché que, en conjunto, ejecutan una instancia de software del motor de Valkey, Memcached o Redis OSS. Al crear un clúster, debe especificar el motor y la versión que se va a utilizar en todos los nodos.

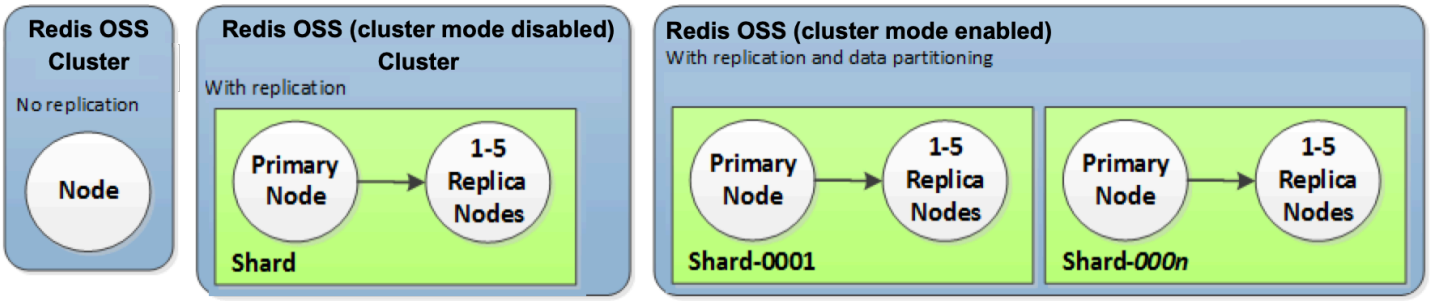
Clústeres de Valkey y Redis OSS

En el siguiente diagrama, se ilustra un clúster típico de Valkey o Redis OSS. Estos clústeres pueden contener un solo nodo o hasta seis nodos dentro de una partición (API/CLI: grupo de nodos). Los clústeres de Valkey o Redis OSS con un solo nodo (modo de clúster deshabilitado) no tienen particiones, mientras que los clústeres de Valkey o Redis OSS (modo de clúster deshabilitado) con varios nodos tienen una única partición. Los clústeres de Valkey o Redis OSS (modo de clúster habilitado) pueden tener hasta 500 particiones, con sus datos divididos entre las particiones. El límite de nodos o particiones se puede aumentar a un máximo de 500 por clúster si la versión del motor es Valkey 7.2 y posteriores o Redis OSS 5.0.6 y posteriores. Por ejemplo, puede elegir configurar un clúster de 500 nodos que oscila entre 83 particiones (uno primario y 5 réplicas por partición) y 500 particiones (único primario y sin réplicas). Asegúrese de que hay suficientes direcciones IP disponibles para acomodar el aumento. Algunos problemas comunes incluyen que las subredes del grupo de subredes tienen un rango CIDR demasiado pequeño o que otros clústeres comparten y utilizan considerablemente las subredes. Para obtener más información, consulte Creación de un grupo de subredes. Para las versiones inferiores a 5.0.6, el límite es de 250 por clúster.

Para solicitar un aumento del límite, consulte AWS Service Limits y elija el tipo de límite Nodes per cluster per instance type (Nodos por clúster por tipo de instancias).

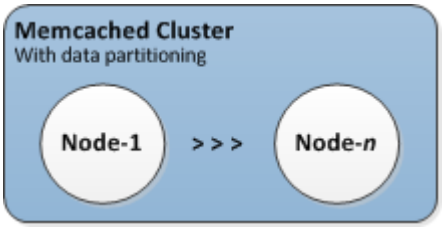
Si tiene varios nodos en una partición de Valkey o Redis OSS, uno de los nodos será un nodo principal de lectura/escritura. Todos los demás nodos de la partición son réplicas de solo lectura.

Los clústeres de Valkey o Redis OSS típicos son similares a los que se muestran a continuación.



Clústeres de Memcached

Los clústeres de Memcached típicos son similares a los que se muestran a continuación. Los clústeres de Memcached contienen entre 1 y 60 nodos entre los que se particionan los datos horizontalmente.



Operaciones de ElastiCache para Valkey, Memcached y Redis OSS

La mayoría de las operaciones de ElastiCache se realizan en el clúster. Puede configurar un clúster con un número específico de nodos y un grupo de parámetros que controle las propiedades de cada nodo. Todos los nodos de un clúster están diseñados para ser del mismo tipo y tener los mismos valores de configuración de parámetros y grupo de seguridad.

Cada clúster debe tener un identificador de clúster. El identificador del clúster es un nombre suministrado por el cliente para el clúster. Este identificador especifica un clúster determinado al interactuar con los comandos de la AWS CLI y la API de ElastiCache. El identificador del clúster debe ser único para ese cliente en una región de AWS.

ElastiCache admite varias versiones del motor. A menos que tenga motivos específicos, recomendamos que utilice la versión más reciente.

Los clústeres de ElastiCache se han diseñado para que se tenga acceso a ellos mediante una instancia de Amazon EC2. Si el clúster se lanza en una nube virtual privada (VPC) en función del servicio de Amazon VPC, se puede acceder a él desde fuera de AWS. Para obtener más información, consulte Acceso a recursos de ElastiCache desde fuera de AWS.

Para obtener una lista de las versiones compatibles, consulte Motores y versiones compatibles, Versiones compatibles del motor de Redis OSS y Versiones compatibles de ElastiCache para Memcached.

Elección de un tipo de red en ElastiCache

ElastiCache admite las versiones 4 y 6 del Protocolo de Internet (IPv4 e IPv6), lo que le permite configurar el clúster para aceptar:

- solo conexiones IPv4,
- solo conexiones IPv6,
- conexiones IPv4 e IPv6 (doble pila).

IPv6 se admite en cargas de trabajo que utilicen Valkey 7.2 y versiones posteriores o Redis OSS versión 6.2 y posteriores en todas las instancias creadas en Nitro System. No hay cargos adicionales por acceder a ElastiCache a través de IPv6.

> **Note**
>
> No se admite la migración de clústeres creados antes de la disponibilidad de IPV6/doble pila. Tampoco se admite el cambio entre tipos de red en los clústeres recién creados.

Se admite IPv6 para cargas de trabajo que utilicen la versión 1.6.6 o superiores de Memcached en todas las instancias creadas en Nitro System. No hay cargos adicionales por acceder a ElastiCache a través de IPv6.

Configuración de subredes para el tipo de red

Si crea un clúster en una Amazon VPC, debe especificar un grupo de subredes. ElastiCache utiliza dicho grupo de subredes para elegir una subred y direcciones IP pertenecientes a dicha subred a fin de asociarlas a sus nodos. Los clústeres de ElastiCache requieren una subred de doble pila con direcciones IPv4 e IPv6 asignadas para funcionar en modo de doble pila y una subred solo IPv6 para funcionar como solo IPv6.

Usar doble pila

Cuando se utiliza ElastiCache para Redis OSS en modo de clúster habilitado, desde la perspectiva de una aplicación, conectarse a todos los nodos del clúster a través del punto de conexión de la configuración no es diferente a conectarse directamente a un nodo de caché individual. Para lograr esto, un cliente que tiene en cuenta los clústeres debe interactuar en un proceso de detección de

clústeres y solicitar la información de configuración de todos los nodos. El protocolo de detección de Redis solo admite una IP por nodo.

Al crear un clúster con ElastiCache para Memcached y elegir la doble pila como tipo de red, debe designar un tipo de detección de IP: IPv4 o IPv6. ElastiCache establecerá de forma predeterminada el tipo de red y la detección de IP en IPv6, pero eso se puede cambiar. Si utiliza la detección automática, solo las direcciones IP del tipo de IP elegido se devolverán al cliente de Memcached. Para obtener más información, consulte Identificación automática de los nodos en el clúster (Memcached).

Para mantener la compatibilidad con versiones anteriores de todos los clientes existentes, se usa la detección de IP, que permite seleccionar el tipo de IP (es decir, IPv4 o IPv6) que se anunciará en el protocolo de detección. Aunque esto limita la detección automática a un solo tipo de IP, la doble pila sigue siendo beneficiosa para las cargas de trabajo habilitadas en modo de clúster, ya que permite migraciones (o retrocesos) de un tipo de IP de detección IPv4 a uno IPv6 sin tiempo de inactividad.

Clústeres de ElastiCache de doble pila compatibles con TLS

Cuando se habilita TLS para los clústeres de ElastiCache, las funciones de detección de clústeres como `cluster slots`, `cluster shards` y `cluster nodes` con Valkey o Redis OSS y `config get cluster` con Memcached devuelven nombres de host en lugar de IP. A continuación, se utilizan los nombres de host en lugar de las IP para conectarse al clúster de ElastiCache y realizar un protocolo de enlace TLS. Esto significa que los clientes no se verán afectados por el parámetro de detección de IP. En el caso de los clústeres habilitados para TLS, el parámetro de detección de IP no tiene ningún efecto en el protocolo IP preferido. En cambio, el protocolo IP utilizado se determinará según el protocolo IP que prefiera el cliente al resolver los nombres de host de DNS.

Para ver ejemplos sobre cómo configurar una preferencia de protocolo IP al resolver nombres de host DNS, consulte Clústeres de ElastiCache de doble pila compatibles con TLS.

Uso de la Consola de administración de AWS (OSS Valkey y Redis OSS)

Al crear un clúster mediante la Consola de administración de AWS, en Connectivity (Conectividad), elija un tipo de red, IPv4, IPv6 o Dual stack (Doble pila). Si está creando un clúster de Valkey o Redis OSS (modo de clúster habilitado) y elige la doble pila, debe seleccionar un Tipo de IP de detección, que puede ser IPv6 o IPv4.

Para obtener más información, consulte Creación de un clúster de Valkey o Redis OSS (modo de clúster habilitado) (consola) o Creación un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) (consola).

Al crear un grupo de replicación mediante la Consola de administración de AWS, elija un tipo de red: IPv4, IPv6 o Dual stack (Doble pila). Si elige doble pila, debe seleccionar un Discovery IP type (Tipo de IP de detección) IPv6 o IPv4.

Para obtener más información, consulte Creación de un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) desde cero o Creación de un grupo de replicación en Valkey o Redis OSS (modo de clúster habilitado) desde cero.

Uso de la Consola de administración de AWS (Memcached)

Al crear un clúster mediante la Consola de administración de AWS, en Connectivity (Conectividad), elija un tipo de red, IPv4, IPv6 o Dual stack (Doble pila). Si elige la doble pila, debe seleccionar un Discovery IP type (Tipo de IP de detección) IPv6 o IPv4.

Para obtener más información, consulte Creación de un clúster de Memcached (consola).

Uso de la CLI con Valkey, Memcached o Redis OSS.

Redis OSS

Al crear un clúster con Valkey o Redis OSS desde la CLI, se utiliza el comando create-cache-cluster y se especifican los parámetros `NetworkType` y `IPDiscovery`:

Para Linux, macOS o Unix:

```
aws elasticache create-cache-cluster \
    --cache-cluster-id "cluster-test" \
    --engine redis \
    --cache-node-type cache.m5.large \
    --num-cache-nodes 1 \
    --network-type dual_stack \
    --ip-discovery ipv4
```

Para Windows:

```
aws elasticache create-cache-cluster ^
    --cache-cluster-id "cluster-test" ^
    --engine redis ^
    --cache-node-type cache.m5.large ^
    --num-cache-nodes 1 ^
    --network-type dual_stack ^
```

```
    --ip-discovery ipv4
```

Al crear un grupo de replicación con el modo de clúster desactivado mediante la CLI, se utiliza el comando create-replication-group y se especifican los parámetros `NetworkType` y `IPDiscovery`:

Para Linux, macOS o Unix:

```
aws elasticache create-replication-group \
   --replication-group-id sample-repl-group \
   --replication-group-description "demo cluster with replicas" \
   --num-cache-clusters 3 \
   --primary-cluster-id redis01 \
   --network-type dual_stack \
   --ip-discovery ipv4
```

Para Windows:

```
aws elasticache create-replication-group ^
   --replication-group-id sample-repl-group ^
   --replication-group-description "demo cluster with replicas" ^
   --num-cache-clusters 3 ^
   --primary-cluster-id redis01 ^
   --network-type dual_stack ^
   --ip-discovery ipv4
```

Al crear un grupo de replicación con el modo de clúster habilitado y usar IPv4 para detección de IP mediante la CLI, se utiliza el comando create-replication-group y se especifican los parámetros `NetworkType` y `IPDiscovery`:

Para Linux, macOS o Unix:

```
aws elasticache create-replication-group \
  --replication-group-id demo-cluster \
  --replication-group-description "demo cluster" \
  --cache-node-type cache.m5.large \
  --num-node-groups 2 \
  --engine redis \
  --cache-subnet-group-name xyz \
  --network-type dual_stack \
  --ip-discovery ipv4 \
```

```
  --region us-east-1
```

Para Windows:

```
aws elasticache create-replication-group ^
  --replication-group-id demo-cluster ^
  --replication-group-description "demo cluster" ^
  --cache-node-type cache.m5.large ^
  --num-node-groups 2 ^
  --engine redis ^
  --cache-subnet-group-name xyz ^
  --network-type dual_stack ^
  --ip-discovery ipv4 ^
  --region us-east-1
```

Al crear un grupo de replicación con el modo de clúster habilitado y usar IPv6 para detección de IP mediante la CLI, se utiliza el comando create-replication-group y se especifican los parámetros `NetworkType` y `IPDiscovery`:

Para Linux, macOS o Unix:

```
aws elasticache create-replication-group \
  --replication-group-id demo-cluster \
  --replication-group-description "demo cluster" \
  --cache-node-type cache.m5.large \
  --num-node-groups 2 \
  --engine redis \
  --cache-subnet-group-name xyz \
  --network-type dual_stack \
  --ip-discovery ipv6 \
  --region us-east-1
```

Para Windows:

```
aws elasticache create-replication-group ^
  --replication-group-id demo-cluster ^
  --replication-group-description "demo cluster" ^
  --cache-node-type cache.m5.large ^
  --num-node-groups 2 ^
  --engine redis ^
  --cache-subnet-group-name xyz ^
  --network-type dual_stack ^
```

```
  --ip-discovery ipv6 ^
  --region us-east-1
```

Memcached

Al crear un clúster con Memcached mediante la CLI, se utiliza el comando create-cache-cluster y se especifican los parámetros `NetworkType` y `IPDiscovery`:

Para Linux, macOS o Unix:

```
aws elasticache create-cache-cluster \
    --cache-cluster-id "cluster-test" \
    --engine memcached \
    --cache-node-type cache.m5.large \
    --num-cache-nodes 1 \
    --network-type dual_stack \
    --ip-discovery ipv4
```

Para Windows:

```
aws elasticache create-cache-cluster ^
    --cache-cluster-id "cluster-test" ^
    --engine memcached ^
    --cache-node-type cache.m5.large ^
    --num-cache-nodes 1 ^
    --network-type dual_stack ^
    --ip-discovery ipv4
```

Identificación automática de los nodos en el clúster (Memcached)

Para los clústeres que ejecutan el motor de Memcached, ElastiCache es compatible con la detección automática, capacidad que permite a los programas de cliente identificar automáticamente todos los nodos de un clúster e iniciar y mantener las conexiones a todos estos nodos.

> **Note**
>
> La detección automática se añade a los clústeres que se ejecutan en Amazon ElastiCache para Memcached. La detección automática no está disponible para los motores de Valkey o Redis OSS.

Con la función Auto Discovery, su aplicación no necesita conectarse manualmente a los nodos de caché individuales; en su lugar, la aplicación se conecta a un nodo de Memcached y recupera la lista de nodos. Con esa lista, su aplicación identificará los demás nodos del clúster y se podrá conectar a cualquiera de ellos. Por lo tanto, no tendrá que codificar de forma rígida los puntos de enlace de nodo de caché en su aplicación.

Si utiliza el tipo de red de pila doble en el clúster, la detección automática solo devolverá direcciones IPv4 o IPv6, según la que seleccione. Para obtener más información, consulte Elección de un tipo de red en ElastiCache .

Todos los nodos de caché del clúster mantienen una lista de metadatos de todos los demás nodos. Estos metadatos se actualizan cuando se agregan o se borran nodos del clúster.

Temas

- Ventajas de la detección automática con Memcached
- Funcionamiento de la detección automática
- Uso de la detección automática
- Conexión manual a los nodos de caché de Memcached
- Adición de detección automática a la biblioteca cliente de Memcached
- Clientes de ElastiCache con detección automática

Ventajas de la detección automática con Memcached

La detección automática tiene las siguientes ventajas cuando se utiliza con Memcached:

- Al aumentar el número de nodos en un clúster, los nuevos nodos se registran con el punto de conexión de configuración y con todos los demás nodos. Al eliminar nodos del clúster de caché, los nodos salientes cancelan su registro. En ambos casos, todos los demás nodos del clúster se actualizan con los metadatos de nodos de caché más recientes.
- Los errores de nodos de caché se detectan automáticamente y los nodos que presentan errores se reemplazan automáticamente.

  > **Note**
  >
  > Los nodos con error seguirán dando error hasta que se complete su reemplazo.

- Los programas cliente solo necesitan conectarse al punto de enlace de configuración. Tras realizar la conexión, la biblioteca de la función Auto Discovery se conecta a todos los demás nodos del clúster.
- Los programas cliente sondean el clúster una vez por minuto (este intervalo puede ajustarse si es necesario). Si se producen cambios en la configuración del clúster como, por ejemplo, al agregar nuevos nodos o eliminar nodos, el cliente recibe una lista actualizada de metadatos. A continuación, el cliente se conecta a estos nodos o se desconecta de ellos según sea necesario.

La detección automática está activada en todos los clústeres de ElastiCache Memcached. Por lo tanto, no es necesario reiniciar ninguno de los nodos de caché para usar esta funcionalidad.

Funcionamiento de la detección automática

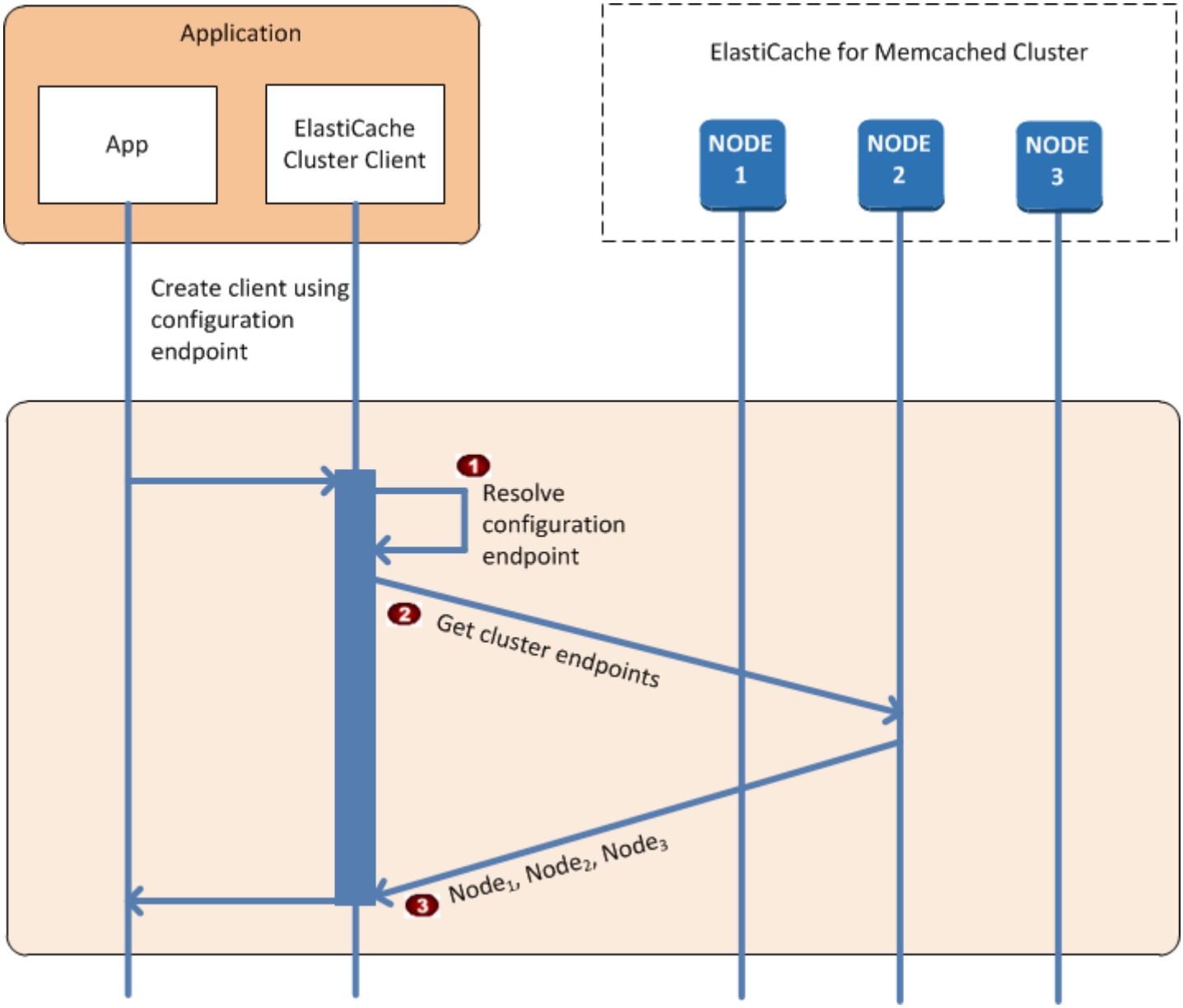
Temas

- Conexión a los nodos de caché
- Operaciones de clúster normales
- Otras operaciones

En esta sección, se describe cómo las aplicaciones cliente utilizan ElastiCache Cluster Client para administrar las conexiones de nodo de caché e interactuar con los elementos de datos de la caché.

Conexión a los nodos de caché

Desde el punto de vista de la aplicación, la conexión al punto de enlace de configuración del clúster no es distinta de la conexión directa a un nodo de caché individual. El siguiente diagrama de secuencias muestra el proceso de enlace a los nodos de caché.



Proceso de enlace a los nodos de caché

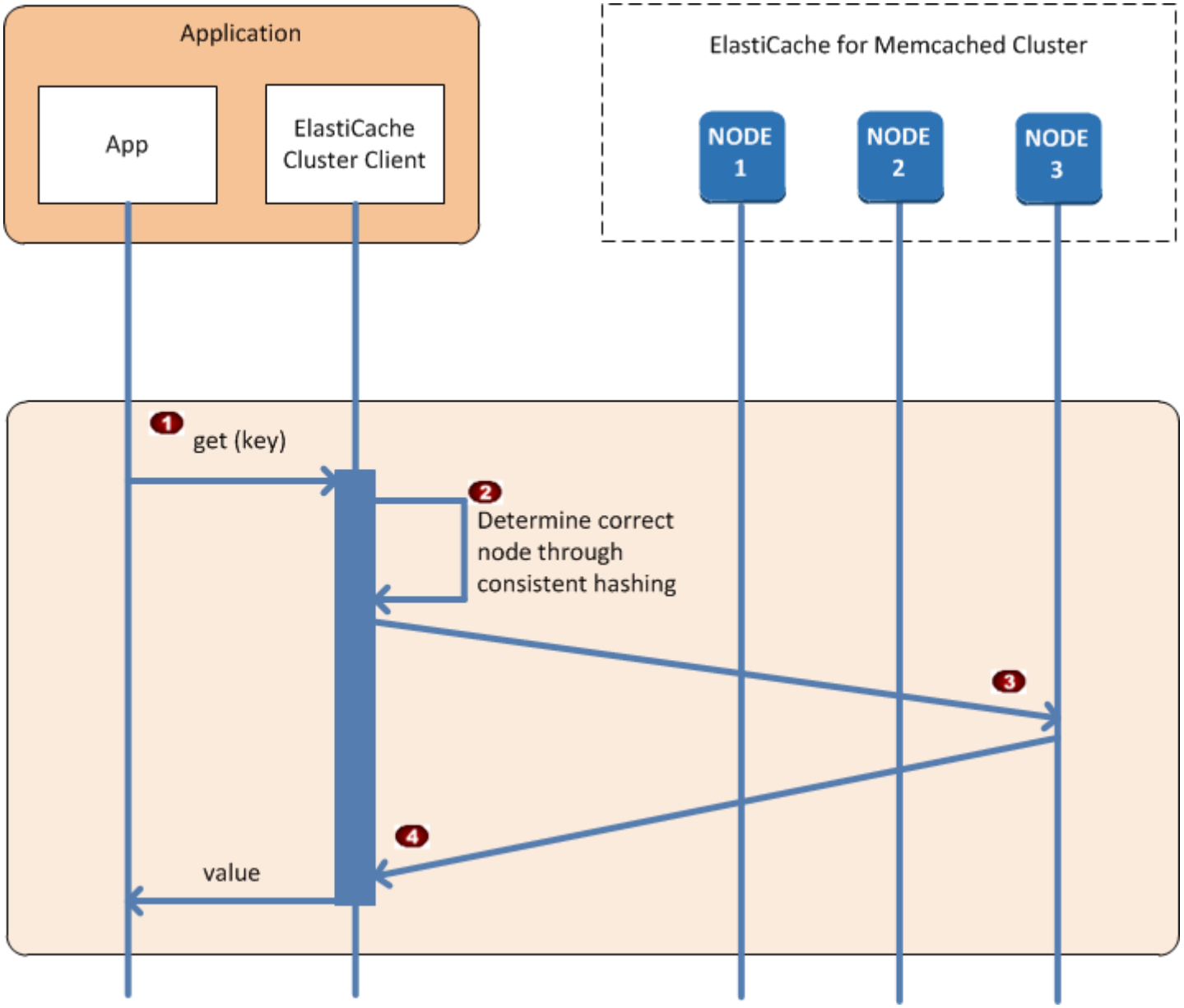
- La aplicación resuelve el nombre de DNS del punto de enlace de configuración. Puesto que el punto de enlace de configuración mantiene las entradas de CNAME para todos los nodos de caché, el nombre de DNS se resuelve en uno de los nodos. De este modo, el cliente puede conectarse a dicho nodo.
- El cliente solicita la información de configuración para todos los demás nodos. Puesto que cada nodo mantiene información de configuración para todos los nodos del clúster, cualquier nodo puede pasar la información de configuración al cliente bajo petición.

- El cliente recibe la lista actual de direcciones IP y nombres de host de los nodos de caché. De este modo, se puede conectar a todos los demás nodos del clúster.

> **Note**
>
> El programa cliente actualiza su lista de direcciones IP y nombres de host de los nodos de caché una vez por minuto. Este intervalo de sondeo se puede ajustar si es necesario.

Operaciones de clúster normales

Cuando la aplicación se ha conectado a todos los nodos de caché, ElastiCache Cluster Client determina qué nodos deben almacenar elementos de datos individuales y a qué nodos es necesario consultar dichos elementos de datos más adelante. El siguiente diagrama de secuencias muestra el proceso de las operaciones de clúster normales.



Proceso de operaciones de clúster normales

- La aplicación emite una solicitud get para obtener un elemento de datos determinado identificado por su clave.
- El cliente usa un algoritmo de hash con la clave para determinar qué nodo de caché contiene el elemento de datos.
- El elemento de datos se solicita desde el nodo adecuado.
- El elemento de datos se devuelve a la aplicación.

Otras operaciones

En algunas situaciones, puede realizar un cambio en los nodos de un clúster. Por ejemplo, puede agregar un nodo adicional para adaptarse a la demanda adicional o eliminar un nodo a fin de ahorrar dinero durante periodos de demanda reducida. O puede reemplazar un nodo debido a un error de nodo de un tipo u otro.

Cuando se produce un cambio en el clúster que requiere actualizar los metadatos con los puntos de enlace del clúster, dicho cambio se realiza en todos los nodos al mismo tiempo. De este modo se consigue que los metadatos de un nodo determinado coincidan con los de los demás nodos del clúster.

En cada uno de estos casos, los metadatos son siempre homogéneos entre todos los nodos, ya que los metadatos se actualizan al mismo tiempo en todos los nodos del clúster. Use siempre el punto de enlace de configuración para obtener los puntos de enlace a los distintos nodos del clúster. Al usar el punto de enlace de configuración, se asegurará de no obtener datos de puntos de enlace de un nodo que “desaparecerá”.

Agregar un nodo

Durante la activación del nodo, su punto de enlace no se incluye en los metadatos. Tan pronto como el nodo esté disponible, el punto de enlace se agrega a los metadatos de cada uno de los nodos del clúster. En este caso, los metadatos son coherentes entre todos los nodos. De este modo, solo podrá interactuar con el nuevo nodo en cuanto esté disponible. Antes de que el nodo esté disponible, no sabrá que existe e interactuará con los nodos del clúster como si el nuevo nodo no existiese.

Eliminación de un nodo

Cuando se elimina un nodo, su punto de enlace se elimina primero de los metadatos. A continuación, el nodo se elimina del clúster. En este caso, los metadatos de todos los nodos son homogéneos, por lo que no contendrán en ningún momento el punto de enlace del nodo que se va a eliminar mientras el nodo no esté disponible. Durante la eliminación del nodo, el punto de enlace no estará incluido en los metadatos y su aplicación solo interactuará con los nodos restantes (n-1), como si el nodo no existiese.

Reemplazo de un nodo

En caso de error en un nodo, ElastiCache elimina dicho nodo y pone en marcha un reemplazo. El proceso de reemplazo tarda unos minutos. Durante este tiempo, los metadatos de todos los nodos

seguirán mostrando el punto de enlace del nodo con error; sin embargo, toda interacción con el nodo generará un error. Por lo tanto, su lógica debe incluir siempre una lógica de reintentos.

Uso de la detección automática

Para empezar a usar la detección automática con ElastiCache para Memcached, siga estos pasos:

- Obtención del punto de conexión de configuración
- Descarga de ElastiCache Cluster Client
- Modificación de su programa de aplicación

Obtención del punto de conexión de configuración

Para conectarse a un clúster, los programas cliente deben conocer el punto de conexión de configuración del clúster. Consulte el tema Búsqueda de puntos de conexión de un clúster (consola) (Memcached)

También puede usar el comando `aws elasticache describe-cache-clusters` con el parámetro `--show-cache-node-info`:

Sea cual sea el método que use para buscar los puntos de enlace del clúster, el punto de conexión de configuración siempre tendrá la cadena .cfg en su dirección.

Example Búsqueda de puntos de enlace con la AWS CLI de ElastiCache

Para Linux, macOS o Unix:

```
aws elasticache describe-cache-clusters \
    --cache-cluster-id mycluster \
    --show-cache-node-info
```

Para Windows:

```
aws elasticache describe-cache-clusters ^
    --cache-cluster-id mycluster ^
    --show-cache-node-info
```

Esta operación produce un resultado similar al siguiente (formato JSON):

```
{
    "CacheClusters": [
        {
            "Engine": "memcached",
            "CacheNodes": [
```

```
                {
                    "CacheNodeId": "0001",
                    "Endpoint": {
                        "Port": 11211,
                        "Address": "mycluster.fnjyzo.cfg.0001.use1.cache.amazonaws.com"
                    },
                    "CacheNodeStatus": "available",
                    "ParameterGroupStatus": "in-sync",
                    "CacheNodeCreateTime": "2016-10-12T21:39:28.001Z",
                    "CustomerAvailabilityZone": "us-east-1e"
                },
                {
                    "CacheNodeId": "0002",
                    "Endpoint": {
                        "Port": 11211,
                        "Address": "mycluster.fnjyzo.cfg.0002.use1.cache.amazonaws.com"
                    },
                    "CacheNodeStatus": "available",
                    "ParameterGroupStatus": "in-sync",
                    "CacheNodeCreateTime": "2016-10-12T21:39:28.001Z",
                    "CustomerAvailabilityZone": "us-east-1a"
                }
            ],
            "CacheParameterGroup": {
                "CacheNodeIdsToReboot": [],
                "CacheParameterGroupName": "default.memcached1.4",
                "ParameterApplyStatus": "in-sync"
            },
            "CacheClusterId": "mycluster",
            "PreferredAvailabilityZone": "Multiple",
            "ConfigurationEndpoint": {
                "Port": 11211,
                "Address": "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com"
            },
            "CacheSecurityGroups": [],
            "CacheClusterCreateTime": "2016-10-12T21:39:28.001Z",
            "AutoMinorVersionUpgrade": true,
            "CacheClusterStatus": "available",
            "NumCacheNodes": 2,
            "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
            "CacheSubnetGroupName": "default",
            "EngineVersion": "1.4.24",
            "PendingModifiedValues": {},
```

```
            "PreferredMaintenanceWindow": "sat:06:00-sat:07:00",
            "CacheNodeType": "cache.r3.large"
        }
    ]
}
```

Descarga de ElastiCache Cluster Client

Para sacar el máximo partido a la detección automática, los programas de cliente deben utilizar ElastiCache Cluster Client. ElastiCache Cluster Client se encuentra disponible para Java, PHP y .NET, y contiene toda la lógica necesaria a fin de detectar todos los nodos de caché y conectarse a ellos.

Para descargar ElastiCache Cluster Client

1. Inicie sesión en la consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. Desde la consola de ElastiCache, elija ElastiCache Cluster Client y, a continuación, Download (Descargar).

El código fuente de ElastiCache Cluster Client para Java se encuentra disponible en https://github.com/amazonwebservices/aws-elasticache-cluster-client-memcached-for-java. Esta biblioteca se basa en el popular cliente Spymemcached. ElastiCache Cluster Client se encuentra sujeto a la licencia de software de Amazon https://aws.amazon.com/asl. Podrá modificar el código fuente de la manera que estime más conveniente. Podrá incluso incorporar el código a bibliotecas de Memcached de código abierto o al código de su propio cliente.

> **Note**
>
> A fin de utilizar ElastiCache Cluster Client para PHP, primero debe instalarlo en su instancia de Amazon EC2. Para obtener más información, consulte Instalación de ElastiCache Cluster Client para .PHP.
> Para un cliente compatible con TLS, descargue el binario con PHP versión 7.4 o superior.
> A fin de utilizar ElastiCache Cluster Client para .NET, primero debe instalarlo en su instancia de Amazon EC2. Para obtener más información, consulte Instalación de ElastiCache Cluster Client para .NET.

Modificación de su programa de aplicación

Modifique su programa de aplicación para que pueda usar la función Auto Discovery. Las siguientes secciones muestran cómo utilizar ElastiCache Cluster Client para Java, PHP y .NET.

> **Important**
>
> Cuando especifique el punto de conexión de la configuración del clúster, asegúrese de que el punto de conexión tenga la cadena ".cfg" en su dirección, tal como se muestra aquí. No use un CNAME o un punto de conexión que no tengan la cadena ".cfg".
>
> ```
> "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";
> ```
>
> Si no especifica de forma explícita el punto de conexión de la configuración del clúster, realizará la configuración de un nodo específico.

Uso de ElastiCache Cluster Client para Java

El siguiente programa indica cómo utilizar ElastiCache Cluster Client para conectarse a un punto de enlace de configuración del clúster y agregar un elemento de datos a la caché. Con la función Auto Discovery, el programa se conectará a todos los nodos del clúster sin necesidad de ninguna intervención adicional.

```
package com.amazon.elasticache;

import java.io.IOException;
import java.net.InetSocketAddress;

// Import the &AWS;-provided library with Auto Discovery support
import net.spy.memcached.MemcachedClient;

public class AutoDiscoveryDemo {

    public static void main(String[] args) throws IOException {

        String configEndpoint = "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";
        Integer clusterPort = 11211;

        MemcachedClient client = new MemcachedClient(
                                 new InetSocketAddress(configEndpoint,
```

```
                                                        clusterPort));
        // The client will connect to the other cache nodes automatically.

        // Store a data item for an hour.
        // The client will decide which cache host will store this item.
        client.set("theKey", 3600, "This is the data value");
    }
}
```

Uso de ElastiCache Cluster Client para PHP

El siguiente programa indica cómo utilizar ElastiCache Cluster Client para conectarse a un punto de conexión de configuración del clúster y agregar un elemento de datos a la caché. Con la función Auto Discovery, el programa se conectará a todos los nodos del clúster sin necesidad de ninguna intervención adicional.

A fin de utilizar ElastiCache Cluster Client para PHP, primero debe instalarlo en su instancia de Amazon EC2. Para obtener más información, consulte Instalación de ElastiCache Cluster Client para .PHP

```
<?php

 /**
  * Sample PHP code to show how to integrate with the Amazon ElastiCache
  * Auto Discovery feature.
  */

  /* Configuration endpoint to use to initialize memcached client.
   * This is only an example.  */
  $server_endpoint = "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";

  /* Port for connecting to the ElastiCache cluster.
   * This is only an example  */
  $server_port = 11211;

 /**
  * The following will initialize a Memcached client to utilize the Auto Discovery
 feature.
  *
  * By configuring the client with the Dynamic client mode with single endpoint, the
  * client will periodically use the configuration endpoint to retrieve the current
 cache
```

```
 * cluster configuration. This allows scaling the cluster up or down in number of
nodes
 * without requiring any changes to the PHP application.
 *
 * By default the Memcached instances are destroyed at the end of the request.
 * To create an instance that persists between requests,
 *    use persistent_id to specify a unique ID for the instance.
 * All instances created with the same persistent_id will share the same connection.
 * See http://php.net/manual/en/memcached.construct.php for more information.
 */
 $dynamic_client = new Memcached('persistent-id');
 $dynamic_client->setOption(Memcached::OPT_CLIENT_MODE,
Memcached::DYNAMIC_CLIENT_MODE);
 $dynamic_client->addServer($server_endpoint, $server_port);

 /**
 * Store the data for 60 seconds in the cluster.
 * The client will decide which cache host will store this item.
 */
 $dynamic_client->set('key', 'value', 60);


/**
 * Configuring the client with Static client mode disables the usage of Auto Discovery
 * and the client operates as it did before the introduction of Auto Discovery.
 * The user can then add a list of server endpoints.
 */
 $static_client = new Memcached('persistent-id');
 $static_client->setOption(Memcached::OPT_CLIENT_MODE, Memcached::STATIC_CLIENT_MODE);
 $static_client->addServer($server_endpoint, $server_port);

/**
 * Store the data without expiration.
 * The client will decide which cache host will store this item.
 */
 $static_client->set('key', 'value');
 ?>
```

Para ver un ejemplo sobre cómo utilizar ElastiCache Cluster Client con TLS habilitado, consulte Using in transit encryption with PHP and Memcached.

Uso de ElastiCache Cluster Client para .NET

> **Note**
>
> El cliente de clúster .NET de ElastiCache quedó obsoleto en mayo de 2022.

El cliente .NET para ElastiCache es un programa de código abierto que se encuentra disponible en https://github.com/awslabs/elasticache-cluster-config-net.

Las aplicaciones .NET suelen recibir sus configuraciones de los archivos de configuración. A continuación se muestra un archivo de configuración de ejemplo.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
    <configSections>
        <section
            name="clusterclient"
            type="Amazon.ElastiCacheCluster.ClusterConfigSettings,
 Amazon.ElastiCacheCluster" />
    </configSections>

    <clusterclient>
        <!-- the hostname and port values are from step 1 above -->
        <endpoint hostname="mycluster.fnjyzo.cfg.use1.cache.amazonaws.com"
 port="11211" />
    </clusterclient>
</configuration>
```

El programa C# indica cómo utilizar ElastiCache Cluster Client para conectarse a un punto de conexión de configuración del clúster y agregar un elemento de datos a la caché. Con la función Auto Discovery, el programa se conectará a todos los nodos del clúster sin necesidad de ninguna intervención adicional.

```
// ****************
// Sample C# code to show how to integrate with the Amazon ElastiCcache Auto Discovery
 feature.

using System;

using Amazon.ElastiCacheCluster;
```

```
using Enyim.Caching;
using Enyim.Caching.Memcached;

public class DotNetAutoDiscoveryDemo  {

    public static void Main(String[] args)  {

        // instantiate a new client.
        ElastiCacheClusterConfig config = new ElastiCacheClusterConfig();
        MemcachedClient memClient = new MemcachedClient(config);

        // Store the data for 3600 seconds (1hour) in the cluster.
        // The client will decide which cache host will store this item.
        memClient.Store(StoreMode.Set, 3600, "This is the data value.");

    }  // end Main

}  // end class DotNetAutoDiscoverDemo
```

Conexión manual a los nodos de caché de Memcached

Si su programa cliente no usa la detección automática, puede conectarse manualmente a cada uno de los nodos de caché de Memcached. Este es el comportamiento predeterminado para los clientes de Memcached.

Puede obtener una lista de nombres anfitriones y números de puerto de los nodos de caché en AWS Management Console. También puede usar el comando AWS CLI `aws elasticache describe-cache-clusters` con el parámetro `--show-cache-node-info`.

Example

El fragmento de código Java siguiente muestra cómo conectarse a todos los nodos en un clúster de cuatro nodos:

```
...

ArrayList<String> cacheNodes = new ArrayList<String>(
 Arrays.asList(
     "mycachecluster.fnjyzo.0001.use1.cache.amazonaws.com:11211",
     "mycachecluster.fnjyzo.0002.use1.cache.amazonaws.com:11211",
     "mycachecluster.fnjyzo.0003.use1.cache.amazonaws.com:11211",
     "mycachecluster.fnjyzo.0004.use1.cache.amazonaws.com:11211"));

MemcachedClient cache = new MemcachedClient(AddrUtil.getAddresses(cacheNodes));

...
```

> **Important**
>
> Si escala o reduce su clúster verticalmente agregando o eliminando nodos, deberá actualizar la lista de nodos en el código del cliente.

Adición de detección automática a la biblioteca cliente de Memcached

La información de configuración de la detección automática se almacena de forma redundante en cada nodo del clúster de Memcached. Las aplicaciones cliente pueden consultar cualquier nodo de caché y obtener la información de configuración para todos los nodos del clúster.

El modo en que una aplicación realiza esta consulta depende de la versión del motor de caché:

- Si la versión del motor de caché es la 1.4.14 o superior, use el comando `config`.
- Si la versión del motor de caché es inferior a la 1.4.14, use el comando `get AmazonElastiCache:cluster`.

Las salidas de estos dos comandos son idénticas y se describen en la sección Formato de salida siguiente.

Versión del motor de caché 1.4.14 o superior

Si la versión del motor de caché es la 1.4.14 o superior, use el comando `config`. ElastiCache ha agregado este comando a los protocolos binarios y ASCII de Memcached y dicho comando se implementa en ElastiCache Cluster Client. Si desea usar la función Auto Discovery con otra biblioteca de cliente, dicha biblioteca deberá ampliarse para ser compatible con el comando `config`.

> **Note**
>
> La documentación siguiente pertenece al protocolo ASCII; sin embargo, el comando `config` también admite el código ASCII y binario. Si desea agregar la detección automática mediante el protocolo binario, consulte el código fuente de ElastiCache Cluster Client.

Sintaxis

```
config [sub-command] [key]
```

Opciones

Nombre	Descripción	Obligatorio
`sub-command`	Subcomando que se usa para interactuar con un nodo de caché. Para la función Auto Discovery, el subcomando es `get`.	Sí

Nombre	Descripción	Obligatorio
`key`	Clave en la que se almacena la configuración del clúster. Para la función Auto Discovery, la clave se denomina `cluster`.	Sí

Para obtener la información de configuración del clúster, use el comando siguiente:

```
config get cluster
```

Versión del motor de caché 1.4.14 o inferior

Para obtener la información de configuración del clúster, use el comando siguiente:

```
get AmazonElastiCache:cluster
```

> **Note**
>
> No modifique la clave "AmazonElastiCache: cluster", ya que en esta clave es donde reside la información de configuración del clúster. Si sobrescribe esta clave, la configuración del cliente será incorrecta durante un breve periodo (no más de 15 segundos). Transcurrido dicho periodo, ElastiCache actualizará y corregirá de forma automática la información de configuración.

Formato de salida

Tanto si usa `config get cluster` o `get AmazonElastiCache:cluster`, la respuesta se compone de dos líneas:

- Número de versión de la información de configuración. Cada vez que se agrega o elimina un nodo del clúster, el número de versión aumenta un valor.
- Lista de nodos de caché. Cada nodo de la lista se representa con un grupo hostname|ip-address|port y cada se delimita con un espacio.

Al final de cada línea encontrará un retorno de carro y un carácter de salto de línea (CR + LF). La línea de datos contiene un carácter de salto de línea (LF) y se le agregan el retorno de carro y el

carácter de salto de línea (CR + LF). La línea de la versión de configuración termina con el carácter de salto de línea (LF) sin retorno de carro (CR).

A continuación se muestra cómo se representaría un clúster con tres nodos:

```
configversion\n
hostname|ip-address|port hostname|ip-address|port hostname|ip-address|port\n\r\n
```

Cada nodo se muestra con el CNAME y la dirección IP privada. El CNAME siempre estará presente; si la dirección IP privada no está disponible, esta no se mostrará, aunque sí se imprimirán los caracteres de barra vertical "|".

Example

A continuación se muestra un ejemplo de la carga que se obtiene al consultar la información de configuración:

```
CONFIG cluster 0 136\r\n
12\n
myCluster.pc4ldq.0001.use1.cache.amazonaws.com|10.82.235.120|11211
 myCluster.pc4ldq.0002.use1.cache.amazonaws.com|10.80.249.27|11211\n\r\n
END\r\n
```

> **Note**
>
> - La segunda línea indica que la información de configuración se ha modificado doce veces hasta el momento.
> - En la tercera línea, la lista de nodos se muestra por orden alfabético según el nombre de host. Este orden puede ser distinto del que usa actualmente en su aplicación cliente.

Clientes de ElastiCache con detección automática

Los programas cliente de clúster pueden identificar todos los nodos del clúster que ejecutan el motor de Memcached y conectarse a ellos automáticamente.

Esta sección trata sobre la instalación y configuración de los clientes PHP y .NET de ElastiCache para su uso con la detección automática.

Temas

- Instalación y compilación de clientes de clúster
- Configuración de clientes de ElastiCache

Instalación y compilación de clientes de clúster

En esta sección se aborda la instalación, la configuración y la compilación de clientes de clúster de detección automática PHP y .NET de Amazon ElastiCache.

Temas

- Instalación de ElastiCache Cluster Client para .NET
- Instalación de ElastiCache Cluster Client para .PHP
- Compilación del código fuente de ElastiCache Cluster Client para PHP

Instalación de ElastiCache Cluster Client para .NET

Encontrará el código de ElastiCache Cluster Client para .NET como código abierto en https://github.com/awslabs/elasticache-cluster-config-net.

En esta sección se describe cómo instalar, actualizar y eliminar los componentes .NET de ElastiCache Cluster Client en instancias de Amazon EC2. Para obtener más información acerca de la detección automática, consulte Identificación automática de los nodos en el clúster (Memcached). Para obtener código .NET de muestra para usar el cliente, consulte Uso de ElastiCache Cluster Client para .NET.

Temas

- Instalación de .NET
- Descargar ElastiCache Cluster Client para .NET para ElastiCache
- Instalar ensamblajes de AWS con NuGet

Instalación de .NET

Debe tener .NET 3.5 o posterior instalado a fin de poder utilizar el SDK .NET de AWS para ElastiCache. Si no dispone de .NET 3.5 o posterior, puede descargar e instalar la versión más reciente desde http://www.microsoft.com/net.

Descargar ElastiCache Cluster Client para .NET para ElastiCache

A fin de descargar ElastiCache Cluster Client para .NET para ElastiCache

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, haga clic en ElastiCache Cluster Client (Cliente de clúster ElastiCache).
3. En la lista Download ElastiCache Memcached Cluster Client (Descargar cliente de clíuster de ElastiCache Memcached), seleccione .NET y, a continuación, haga clic en Download.

Instalar ensamblajes de AWS con NuGet

NuGet es un sistema de administración de paquetes de la plataforma .NET. NuGet tiene en cuenta las dependencias de los ensamblajes e instala todos los archivos necesarios de forma automática. Los ensamblajes de NuGet instalados se almacenan con su solución, en lugar de en una ubicación central como, por ejemplo, `Program Files`. De este modo, podrá instalar versiones específicas para una aplicación sin crear problemas de compatibilidad.

Instalación de NuGet

NuGet puede instalarse desde la Galería de instalación de MSDN. Para ello, consulte https://visualstudiogallery.msdn.microsoft.com/27077b70-9dad-4c64-adcf-c7cf6bc9970c. Si usa Visual Studio 2010 o posterior, NuGet se instalará automáticamente.

Puede usar NuGet desde el Explorador de soluciones o la Consola del administrador de paquetes.

Uso de NuGet desde el Explorador de soluciones

Para usar NuGet desde el Explorador de soluciones de Visual Studio 2010

1. Desde el menú Herramientas, seleccione Administrador de paquetes de biblioteca.
2. Haga clic en Consola del administrador de paquetes.

Para usar NuGet desde el Explorador de soluciones de Visual Studio 2012 o Visual Studio 2013

1. Desde el menú Herramientas, seleccione Administrador de paquetes de NuGet.
2. Haga clic en Consola del administrador de paquetes.

Desde la línea de comandos, puede instalar los ensamblajes mediante `Install-Package`, tal y como se muestra a continuación.

```
Install-Package Amazon.ElastiCacheCluster
```

Para ver una página con todos los paquetes disponibles a través de NuGet, como los ensamblajes AWSSDK y AWS.Extensions, consulte el sitio web de NuGet en http://www.nuget.org. La página de cada paquete incluye una línea de comandos de muestra para la instalación del paquete mediante la consola, así como una lista de las versiones anteriores del paquete que están disponibles a través de NuGet.

Para obtener más información acerca de los comandos de la Consola del administrador de paquetes, consulte http://nuget.codeplex.com/wikipage?title=Package%20Manager%20Console%20Command%20Reference%20%28v1.3%29.

Instalación de ElastiCache Cluster Client para .PHP

En esta sección se describe cómo instalar, actualizar y eliminar los componentes PHP de ElastiCache Cluster Client en instancias de Amazon EC2. Para obtener más información acerca de la detección automática, consulte Identificación automática de los nodos en el clúster (Memcached). Para obtener código PHP de muestra para usar el cliente, consulte Uso de ElastiCache Cluster Client para PHP.

Temas

- Descarga del paquete de instalación
- Para los usuarios que ya tengan instalada la extensión php-memcached
- Pasos de instalación para usuarios nuevos
- Eliminación de Cluster Client para PHP

Descarga del paquete de instalación

A fin de garantizar que utiliza la versión correcta de ElastiCache Cluster Client para PHP, tendrá que saber qué versión de PHP tiene instalada en su instancia de Amazon EC2. También necesitará saber si su instancia de Amazon EC2 ejecuta una versión de Linux de 64 o 32 bits.

Para determinar la versión de PHP que tiene instalada en su instancia de Amazon EC2

- En el símbolo del sistema, ejecute el siguiente comando:

  ```
  php -v
  ```

  La versión de PHP se mostrará en la salida, como en el ejemplo siguiente:

  ```
  PHP 5.4.10 (cli) (built: Jan 11 2013 14:48:57)
  Copyright (c) 1997-2012 The PHP Group
  Zend Engine v2.4.0, Copyright (c) 1998-2012 Zend Technologies
  ```

  > **Note**
  >
  > Si sus versiones de PHP y de Memcached son incompatibles, se mostrará un mensaje similar al siguiente:
  >
  > ```
  > PHP Warning: PHP Startup: memcached: Unable to initialize module
  > ```

```
Module compiled with module API=20100525
PHP compiled with module API=20131226
These options need to match
in Unknown on line 0
```

Si esto ocurre, deberá compilar el módulo desde el código fuente. Para obtener más información, consulte Compilación del código fuente de ElastiCache Cluster Client para PHP.

Para determinar su arquitectura de AMI de Amazon EC2 (de 64 o 32 bits)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon EC2 en https://console.aws.amazon.com/ec2/.
2. En la lista de Instances (Instancias), haga clic en su instancia de Amazon EC2.
3. En la pestaña Description, busque el campo AMI:. Una instancia de 64 bits debe tener `x86_64` como parte de la descripción, mientras que la instancia de 32 bits deberá tener el valor `i386` o `i686` en este campo.

Ahora ya puede descargar ElastiCache Cluster Client.

A fin de descargar ElastiCache Cluster Client para PHP

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En la consola de ElastiCache, elija ElastiCache Cluster Client.
3. Desde la lista de Download ElastiCache Memcached Cluster Client (Descargar ElastiCache Cluster Client para Memcached), elija el ElastiCache Cluster Client que coincida con su versión de PHP y su arquitectura AMI. A continuación, elija el botón Download (Descargar).

Para los usuarios que ya tengan instalada la extensión php-memcached

Para actualizar la instalación de **`php-memcached`**

1. Elimine la instalación anterior de la extensión de Memcached para PHP tal y como se describe en el tema Eliminación de Cluster Client para PHP.

2. Instale la nueva extensión `php-memcached` de ElastiCache tal y como se ha descrito anteriormente en Pasos de instalación para usuarios nuevos.

Pasos de instalación para usuarios nuevos

Temas

- Instalación de PHP 7.x. para usuarios nuevos
- Instalación de PHP 5.x para usuarios nuevos

Instalación de PHP 7.x. para usuarios nuevos

Temas

- Instalación de PHP 7 en la AMI de Ubuntu Server 14.04 LTS (64 y 32 bits)
- Para instalar PHP 7 AMI de Amazon Linux 201609
- Para instalar PHP 7 en una AMI de SUSE Linux

Instalación de PHP 7 en la AMI de Ubuntu Server 14.04 LTS (64 y 32 bits)

1. Lance una instancia nueva desde la AMI.
2. Ejecute los comandos siguientes:

```
sudo apt-get update
sudo apt-get install gcc g++
```

3. Instale PHP 7.

```
sudo yum install php70
```

4. Descargue Amazon ElastiCache Cluster Client.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.0/
latest-64bit
```

5. EXTRACT `latest-64bit`.

```
tar -zxvf latest-64bit
```

6. Con permisos raíz, copie el archivo de artefacto extraído `amazon-elasticache-cluster-client.so` a `/usr/lib/php/20151012`.

```
sudo mv artifact/amazon-elasticache-cluster-client.so /usr/lib/php/20151012
```

7. Inserte la línea `extension=amazon-elasticache-cluster-client.so` en el archivo `/etc/php/7.0/cli/php.ini`.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/
php/7.0/cli/php.ini
```

8. Inicie o reinicie su servidor de Apache.

```
sudo /etc/init.d/httpd start
```

Para instalar PHP 7 AMI de Amazon Linux 201609

1. Lance una instancia nueva desde la AMI.
2. Ejecuta el siguiente comando:

```
sudo yum install gcc-c++
```

3. Instale PHP 7.

```
sudo yum install php70
```

4. Descargue Amazon ElastiCache Cluster Client.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.0/
latest-64bit
```

5. EXTRACT `latest-64bit`.

```
tar -zxvf latest-64bit
```

6. Con permisos raíz, copie el archivo de artefacto extraído `amazon-elasticache-cluster-client.so` a `/usr/lib64/php/7.0/modules/`.

```
sudo mv artifact/amazon-elasticache-cluster-client.so /usr/lib64/php/7.0/modules/
```

7. Creación del archivo `50-memcached.ini`.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/
php-7.0.d/50-memcached.ini
```

8. Inicie o reinicie su servidor de Apache.

```
sudo /etc/init.d/httpd start
```

Para instalar PHP 7 en una AMI de SUSE Linux

1. Lance una instancia nueva desde la AMI.
2. Ejecuta el siguiente comando:

```
sudo zypper install gcc
```

3. Instale PHP 7.

```
sudo yum install php70
```

4. Descargue Amazon ElastiCache Cluster Client.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.0/
latest-64bit
```

5. EXTRACT `latest-64bit`.

```
tar -zxvf latest-64bit
```

6. Con permisos raíz, copie el archivo de artefacto extraído `amazon-elasticache-cluster-client.so` a `/usr/lib64/php7/extensions/`.

```
sudo mv artifact/amazon-elasticache-cluster-client.so /usr/lib64/php7/extensions/
```

7. Inserte la línea `extension=amazon-elasticache-cluster-client.so` en el archivo `/etc/php7/cli/php.ini`.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/
php7/cli/php.ini
```

8. Inicie o reinicie su servidor de Apache.

```
sudo /etc/init.d/httpd start
```

Instalación de PHP 5.x para usuarios nuevos

Temas

- Para instalar PHP 5 en una AMI de Amazon Linux 2014.03 (64 y 32 bits)
- Para instalar PHP 5 en una AMI de Red Hat Enterprise Linux 7.0 (64 y 32 bits)
- Para instalar PHP 5 en la AMI de Ubuntu Server 14.04 LTS (64 y 32 bits)
- A fin de instalar PHP 5 para la AMI de SUSE Linux Enterprise Server 11 (64 o 32 bits)
- Otras distribuciones de Linux

Para instalar PHP 5 en una AMI de Amazon Linux 2014.03 (64 y 32 bits)

1. Lance una instancia de Amazon Linux (de 64 o 32 bits) e inicie sesión en ella.
2. Instale las dependencias de PHP:

```
sudo yum install gcc-c++ php php-pear
```

3. Descargue el paquete `php-memcached` correcto para su instancia de Amazon EC2 y su versión de PHP. Para obtener más información, consulte Descarga del paquete de instalación.
4. Instalar `php-memcached`. El URI debe ser la ruta de descarga del paquete de instalación:

```
sudo pecl install <package download path>
```

A continuación se muestra un ejemplo de comando de instalación para PHP 5.4, Linux de 64 bits. En este ejemplo, reemplace *X.Y.Z* con el número de versión real:

```
sudo pecl install /home/AmazonElastiCacheClusterClient-X.Y.Z-PHP54-64bit.tgz
```

> **Note**
>
> Asegúrese de que utiliza la versión más reciente del artefacto de instalación.

5. Con el permiso root/sudo, agregue un nuevo archivo denominado `memcached.ini` en el directorio `/etc/php.d` e inserte “extension=amazon-elasticache-cluster-client.so” en el archivo:

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/
php.d/memcached.ini
```

6. Inicie o reinicie su servidor de Apache.

```
sudo /etc/init.d/httpd start
```

Para instalar PHP 5 en una AMI de Red Hat Enterprise Linux 7.0 (64 y 32 bits)

1. Lance una instancia de Red Hat Enterprise Linux (de 64 o 32 bits) e inicie sesión en ella.
2. Instale las dependencias de PHP:

```
sudo yum install gcc-c++ php php-pear
```

3. Descargue el paquete `php-memcached` correcto para su instancia de Amazon EC2 y su versión de PHP. Para obtener más información, consulte Descarga del paquete de instalación.
4. Instalar `php-memcached`. El URI debe ser la ruta de descarga del paquete de instalación:

```
sudo pecl install <package download path>
```

5. Con el permiso root/sudo, añada un nuevo archivo denominado `memcached.ini` en el directorio `/etc/php.d` e inserte `extension=amazon-elasticache-cluster-client.so` en el archivo.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/
php.d/memcached.ini
```

6. Inicie o reinicie su servidor de Apache.

```
sudo /etc/init.d/httpd start
```

Para instalar PHP 5 en la AMI de Ubuntu Server 14.04 LTS (64 y 32 bits)

1. Lance una instancia de Ubuntu Linux (de 64 o 32 bits) e inicie sesión en ella.
2. Instale las dependencias de PHP:

```
sudo apt-get update
sudo apt-get install gcc g++ php5 php-pear
```

3. Descargue el paquete `php-memcached` correcto para su instancia de Amazon EC2 y su versión de PHP. Para obtener más información, consulte Descarga del paquete de instalación.
4. Instalar `php-memcached`. El URI debe ser la ruta de descarga del paquete de instalación.

```
sudo pecl install <package download path>
```

> **Note**
>
> Este paso de instalación instala el artefacto de compilación `amazon-elasticache-cluster-client.so` en el directorio `/usr/lib/php5/20121212*`. Compruebe la ruta completa del artefacto de compilación, ya que la necesitará en el siguiente paso.

Si el comando anterior no funciona, deberá extraer manualmente el artefacto del cliente de PHP `amazon-elasticache-cluster-client.so` del archivo `*.tgz` descargado y copiarlo en el directorio `/usr/lib/php5/20121212*`.

```
tar -xvf <package download path>
cp amazon-elasticache-cluster-client.so /usr/lib/php5/20121212/
```

5. Con el permiso root/sudo, agregue un nuevo archivo denominado `memcached.ini` en el directorio `/etc/php5/cli/conf.d` e inserte “extension=<absolute path to amazon-elasticache-cluster-client.so>” en el archivo.

```
echo "extension=<absolute path to amazon-elasticache-cluster-client.so>" | sudo tee
 --append /etc/php5/cli/conf.d/memcached.ini
```

6. Inicie o reinicie su servidor de Apache.

```
sudo /etc/init.d/httpd start
```

A fin de instalar PHP 5 para la AMI de SUSE Linux Enterprise Server 11 (64 o 32 bits)

1. Lance una instancia de SUSE Linux (de 64 o 32 bits) e inicie sesión en ella.
2. Instale las dependencias de PHP:

```
sudo zypper install gcc php53-devel
```

3. Descargue el paquete `php-memcached` correcto para su instancia de Amazon EC2 y su versión de PHP. Para obtener más información, consulte Descarga del paquete de instalación.
4. Instalar `php-memcached`. El URI debe ser la ruta de descarga del paquete de instalación.

```
sudo pecl install <package download path>
```

5. Con el permiso root/sudo, añada un nuevo archivo denominado `memcached.ini` en el directorio `/etc/php5/conf.d` e inserte **`extension=amazon-elasticache-cluster-client.so`** en el archivo.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/
php5/conf.d/memcached.ini
```

6. Inicie o reinicie su servidor de Apache.

```
sudo /etc/init.d/httpd start
```

> **Note**
>
> Si el paso 5 no funciona en ninguna de las plataformas anteriores, compruebe la ruta de instalación de `amazon-elasticache-cluster-client.so`. Además, especifique la ruta

completa del archivo binario en la extensión. Compruebe además que el PHP en uso es una versión compatible. Se admiten las versiones de la 5.3 a la 5.5.

Otras distribuciones de Linux

En algunos sistemas, especialmente en CentOS7 y Red Hat Enterprise Linux (RHEL) 7.1, `libsasl2.so.3` ha reemplazado a `libsasl2.so.2`. En estos sistemas, al cargar el cliente del clúster de ElastiCache, este intenta buscar y cargar , pero obtiene un error `libsasl2.so.2`. Para solucionar este problema, cree un enlace a simbólico a `libsasl2.so.3` para que cuando el cliente intente cargar libsasl2.so.2, se redirija a `libsasl2.so.3`. El código siguiente crea este enlace simbólico.

```
cd /usr/lib64
sudo ln libsasl2.so.3 libsasl2.so.2
```

Eliminación de Cluster Client para PHP

Temas

- Eliminación de una versión anterior a PHP 7
- Eliminación de una versión anterior a PHP 5

Eliminación de una versión anterior a PHP 7

Para eliminar una versión anterior a PHP 7

1. Elimine el archivo `amazon-elasticache-cluster-client.so` del directorio de biblioteca PHP correspondiente, tal como se indica en las instrucciones de instalación. Consulte la sección correspondiente a su instalación en Para los usuarios que ya tengan instalada la extensión php-memcached.
2. Elimine la línea `extension=amazon-elasticache-cluster-client.so` del archivo `php.ini`.
3. Inicie o reinicie su servidor de Apache.

   ```
   sudo /etc/init.d/httpd start
   ```

Eliminación de una versión anterior a PHP 5

Para eliminar una versión anterior a PHP 5

1. Elimine la extensión `php-memcached`:

```
sudo pecl uninstall __uri/AmazonElastiCacheClusterClient
```

2. Elimine el archivo `memcached.ini` que se añadió en el directorio correspondiente tal como se indica en los pasos de instalación anteriores.

Compilación del código fuente de ElastiCache Cluster Client para PHP

En esta sección se explica cómo obtener y compilar el código fuente de ElastiCache Cluster Client para PHP.

Existen dos paquetes que debe extraer de GitHub y compilar; aws-elasticache-cluster-client-libmemcached y aws-elasticache-cluster-client-memcached-for-php.

Temas

- Compilación de la biblioteca libmemcached
- Compilación del cliente de detección automática de ElastiCache for Memcached para PHP

Compilación de la biblioteca libmemcached

Para compilar la biblioteca aws-elasticache-cluster-client-libmemcached

1. Lance una instancia de Amazon EC2.
2. Instale las dependencias de la biblioteca.
   - En AMI de Amazon Linux 201509

     ```
     sudo yum install gcc gcc-c++ autoconf libevent-devel
     ```

   - En AMI de Ubuntu 14.04

     ```
     sudo apt-get update
     sudo apt-get install libevent-dev gcc g++ make autoconf libsasl2-dev
     ```

3. Extraiga el repositorio y compile el código.

   ```
   Download and install  https://github.com/awslabs/aws-elasticache-cluster-client-
   libmemcached/archive/v1.0.18.tar.gz
   ```

Compilación del cliente de detección automática de ElastiCache for Memcached para PHP

En las secciones siguientes se describe cómo compilar el cliente de detección automática de ElastiCache for Memcached.

Temas

- Compilación del cliente de ElastiCache for Memcached para PHP 7
- Compilación del cliente de ElastiCache para Memcached para PHP 5

Compilación del cliente de ElastiCache for Memcached para PHP 7

Ejecute el siguiente conjunto de comandos en el directorio de código.

```
git clone https://github.com/awslabs/aws-elasticache-cluster-client-memcached-for-
php.git
cd aws-elasticache-cluster-client-memcached-for-php
git checkout php7
sudo yum install php70-devel
phpize
./configure --with-libmemcached-dir=<libmemcached-install-directory> --disable-
memcached-sasl
make
make install
```

> **Note**
>
> Puede vincular estáticamente la biblioteca de libmemcached en el binario de PHP para poder transferirla entre varias plataformas de Linux. Para ello, ejecute el siguiente comando antes de `make`:
>
> ```
> sed -i "s#-lmemcached#<libmemcached-install-directory>/lib/libmemcached.a -
> lcrypt -lpthread -lm -lstdc++ -lsasl2#" Makefile
> ```

Compilación del cliente de ElastiCache para Memcached para PHP 5

Compile el `aws-elasticache-cluster-client-memcached-for-php` ejecutando los siguientes comandos en la carpeta `aws-elasticache-cluster-client-memcached-for-php/`.

```
git clone https://github.com/awslabs/aws-elasticache-cluster-client-memcached-for-
php.git
cd aws-elasticache-cluster-client-memcached-for-php
sudo yum install zlib-devel
phpize
```

```
./configure --with-libmemcached-dir=<libmemcached-install-directory>
make
make install
```

Configuración de clientes de ElastiCache

Un clúster de ElastiCache utiliza un protocolo compatible con Valkey, Memcached y Redis OSS. El código, las aplicaciones y las herramientas más populares que se usan hoy en día con el entorno existente funcionarán a la perfección con este servicio.

En esta sección se explican las consideraciones específicas para la conexión a los nodos de caché en ElastiCache.

Temas

- Comandos restringidos
- Búsqueda de números de puertos y puntos de enlace de los nodos
- Conexión para utilizar la detección automática
- Conexión a nodos en un clúster de Valkey o Redis OSS
- Nombres de DNS e IP subyacentes

Comandos restringidos

Para ofrecer una experiencia de servicio administrado, ElastiCache restringe el acceso a determinados comandos específicos del motor de caché que requieren privilegios avanzados. A continuación se detallan los comandos que no están disponibles para los clústeres que ejecutan Redis OSS:

- `bgrewriteaof`
- `bgsave`
- `config`
- `debug`
- `migrate`
- `replicaof`
- `save`
- `slaveof`
- `shutdown`
- `sync`

Búsqueda de números de puertos y puntos de enlace de los nodos

Para conectarse a un nodo de caché, su aplicación necesita conocer el punto de enlace y el número de puerto de dicho nodo.

Búsqueda de números de puertos y puntos de enlace de los nodos (consola)

Para determinar los números de puertos y puntos de enlace de nodos

1. Inicie sesión en la consola de administración de Amazon ElastiCache y elija el motor que se ejecuta en el clúster.

   Se mostrará una lista de todos los clústeres que ejecutan el motor elegido.
2. Continúe con el procedimiento que se describe a continuación en función del motor y la configuración que esté ejecutando.
3. Elija el nombre del clúster que le interese.
4. Localice las columnas Port y Endpoint del nodo que le interese.

Búsqueda de números de puertos y puntos de enlace de nodos de caché (AWS CLI)

Para determinar los números de puertos y puntos de enlace de nodos de caché, use el comando `describe-cache-clusters` con el parámetro `--show-cache-node-info`.

```
aws elasticache describe-cache-clusters --show-cache-node-info
```

Los números de puertos y los nombres DNS completos se encuentran en la sección de punto de enlace de la salida.

Búsqueda de números de puertos y puntos de enlace de nodos de caché (API de ElastiCache)

Para determinar los números de puertos y puntos de enlace de nodos de caché, use la acción `DescribeCacheClusters` con el parámetro `ShowCacheNodeInfo=true`.

Example

```
https://elasticache.us-west-2.amazonaws.com /
    ?Action=DescribeCacheClusters
    &ShowCacheNodeInfo=true
```

```
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20140421T220302Z
    &Version=2014-09-30
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Credential=<credential>
    &X-Amz-Date=20140421T220302Z
    &X-Amz-Expires=20140421T220302Z
    &X-Amz-Signature=<signature>
    &X-Amz-SignedHeaders=Host
```

Conexión para utilizar la detección automática

Si sus aplicaciones usan la función de detección automática, solo necesita conocer el punto de enlace del clúster, más que los puntos de enlace individuales de cada nodo de caché. Para obtener más información, consulte Identificación automática de los nodos en el clúster (Memcached).

> **Note**
>
> En este momento, la función de detección automática solo está disponible para los clústeres que ejecutan Memcached.

Conexión a nodos en un clúster de Valkey o Redis OSS

> **Note**
>
> En este momento, los clústeres (API/CLI: grupos de replicación) que admiten la replicación y las réplicas de lectura solo son compatibles con los clústeres que ejecutan Valkey o Redis OSS.

Para los clústeres, ElastiCache proporciona las interfaces de la API, la CLI y la consola a fin de obtener la información de conexión de los nodos individuales.

Para la actividad de solo lectura, las aplicaciones pueden conectarse a cualquier nodo del clúster. Sin embargo, para la actividad de escritura, recomendamos que sus aplicaciones se conecten al punto de conexión principal (Valkey o Redis OSS [modo de clúster deshabilitado]) o al punto de conexión de configuración (Valkey o Redis OSS [modo de clúster habilitado]) del clúster en lugar de

conectarse directamente a un nodo. De este modo, se asegurará de que sus aplicaciones siempre puedan encontrar el nodo correcto, incluso si decide volver a configurar su clúster promocionando una réplica de lectura a la función principal.

Conexión con los clústeres de un grupo de reproducción (consola)

Para determinar los números de puertos y puntos de enlace

- Consulte el tema, Búsqueda de puntos de conexión de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) (consola).

Conexión con los clústeres de un grupo de reproducción (AWS CLI)

Para determinar los números de puertos y los puntos de enlace de los nodos de caché

Use el comando `describe-replication-groups` con el nombre de su grupo de reproducción:

```
aws elasticache describe-replication-groups redis2x2
```

Este comando debería producir un resultado similar al siguiente:

```
{
    "ReplicationGroups": [
        {
            "Status": "available",
            "Description": "2 shards, 2 nodes (1 + 1 replica)",
            "NodeGroups": [
                {
                    "Status": "available",
                    "Slots": "0-8191",
                    "NodeGroupId": "0001",
                    "NodeGroupMembers": [
                        {
                            "PreferredAvailabilityZone": "us-west-2c",
                            "CacheNodeId": "0001",
                            "CacheClusterId": "redis2x2-0001-001"
                        },
                        {
                            "PreferredAvailabilityZone": "us-west-2a",
                            "CacheNodeId": "0001",
                            "CacheClusterId": "redis2x2-0001-002"
                        }
```

```
                    ]
                },
                {
                    "Status": "available",
                    "Slots": "8192-16383",
                    "NodeGroupId": "0002",
                    "NodeGroupMembers": [
                        {
                            "PreferredAvailabilityZone": "us-west-2b",
                            "CacheNodeId": "0001",
                            "CacheClusterId": "redis2x2-0002-001"
                        },
                        {
                            "PreferredAvailabilityZone": "us-west-2a",
                            "CacheNodeId": "0001",
                            "CacheClusterId": "redis2x2-0002-002"
                        }
                    ]
                }
            ],
            "ConfigurationEndpoint": {
                "Port": 6379,
                "Address": "redis2x2.9dcv5r.clustercfg.usw2.cache.amazonaws.com"
            },
            "ClusterEnabled": true,
            "ReplicationGroupId": "redis2x2",
            "SnapshotRetentionLimit": 1,
            "AutomaticFailover": "enabled",
            "SnapshotWindow": "13:00-14:00",
            "MemberClusters": [
                "redis2x2-0001-001",
                "redis2x2-0001-002",
                "redis2x2-0002-001",
                "redis2x2-0002-002"
            ],
            "CacheNodeType": "cache.m3.medium",
            "PendingModifiedValues": {}
        }
    ]
}
```

Conexión con los clústeres de un grupo de reproducción (API de ElastiCache)

Para determinar los números de puertos y los puntos de enlace de los nodos de caché

Llame a `DescribeReplicationGroups` con el siguiente parámetro:

`ReplicationGroupId` = nombre de su grupo de reproducción.

Example

```
https://elasticache.us-west-2.amazonaws.com /
    ?Action=DescribeCacheClusters
    &ReplicationGroupId=repgroup01
    &Version=2014-09-30
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20140421T220302Z
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20140421T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20140421T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Nombres de DNS e IP subyacentes

Los clientes conservan una lista de servidores con las direcciones y los puertos de los servidores donde se almacenan los datos de la caché. Cuando se utiliza ElastiCache, la API DescribeCacheClusters (o la utilidad de la línea de comandos describe-cache-clusters) devuelve una entrada de DNS completa y un número de puerto que podrá utilizar para la lista de servidores.

> **Important**
>
> Es importante que las aplicaciones cliente estén configuradas para resolver los nombres de DNS con frecuencia de los nodos de caché cuando intenten conectarse a un punto de enlace de nodo de caché.

ElastiCache garantiza que el nombre de DNS de los nodos de caché no se modifiquen al recuperar los nodos de caché en caso de error.

De forma predeterminada, la mayoría de las bibliotecas de cliente admiten las conexiones de nodo de caché persistentes Se recomienda utilizar conexiones de nodo de caché persistentes cuando utilice ElastiCache. El almacenamiento en caché de DNS del lado del cliente puede producirse en

varios lugares, incluidas las bibliotecas de cliente, el tiempo de ejecución del lenguaje o el sistema operativo del cliente. Debe revisar la configuración de su aplicación en cada capa para garantizar que las direcciones IP de sus nodos de caché se resuelven con frecuencia.

Organización de datos en niveles en ElastiCache

Los clústeres de ElastiCache para Valkey o Redis OSS que componen un grupo de replicación y utilizan un tipo de nodo de la familia r6gd tienen sus datos organizados en niveles entre la memoria y el almacenamiento SSD local (unidades de estado sólido). La organización de datos en niveles proporciona una nueva opción de precio-rendimiento para las cargas de trabajo de Valkey o Redis OSS mediante el uso de unidades de estado sólido (SSD) de menor costo en cada nodo de clúster, además de almacenar datos en la memoria. Es ideal para cargas de trabajo que acceden regularmente hasta un 20 por ciento de su conjunto de datos general y para aplicaciones que pueden tolerar latencia adicional al acceder a los datos en SSD.

En clústeres con organización de datos en niveles de ElastiCache, ElastiCache supervisa la última hora de acceso de cada elemento que almacena. Cuando la memoria disponible (DRAM) se consume por completo, ElastiCache utiliza un algoritmo menos usado recientemente (LRU) para mover automáticamente los elementos a los que se accede con poca frecuencia de la memoria a la SSD. Cuando se accede posteriormente a los datos de SSD, ElastiCache los mueve de nuevo a la memoria de forma automática y asincrónica antes de procesar la solicitud. Si tiene una carga de trabajo que solo accede a un subconjunto de sus datos regularmente, la organización de datos en niveles es una forma óptima de escalar su capacidad de forma rentable.

Tenga en cuenta que cuando se utiliza la organización de datos en niveles, las propias claves siempre permanecen en la memoria, mientras que la LRU rige la ubicación de los valores en la memoria frente al disco. En general, recomendamos que los tamaños de las claves sean más pequeños que los tamaños de los valores al usar la organización de datos en niveles.

La organización de datos en niveles está diseñada para tener un impacto mínimo en el rendimiento de las cargas de trabajo Por ejemplo, suponiendo valores de cadena de 500 bytes, puede esperar 300 microsegundos adicionales de latencia en promedio para las solicitudes de datos almacenados en SSD en comparación con las solicitudes de datos de la memoria.

Con el mayor tamaño de nodo de organización de datos en niveles (cache.r6gd.16xlarge), puede almacenar hasta 1 petabyte en un solo clúster de 500 nodos (500 TB cuando se utiliza 1 réplica de lectura). La organización de datos en niveles es compatible con todos los comandos y estructuras de datos de Valkey o Redis OSS compatibles con ElastiCache. No es necesario cambiar el lado del cliente para usar esta característica.

Temas

- Prácticas recomendadas
- Limitaciones
- Precios
- Monitorización
- Uso de la organización de datos en niveles
- Restauración de datos desde copias de seguridad en clústeres con la organización de datos en niveles

Prácticas recomendadas

Recomendamos que siga las siguientes prácticas recomendadas:

- La organización de datos en niveles es ideal para cargas de trabajo que acceden regularmente hasta un 20 % de su conjunto de datos general y para aplicaciones que pueden tolerar latencia adicional al acceder a los datos en SSD.
- Al utilizar la capacidad de SSD disponible en nodos con niveles de datos, recomendamos que el tamaño del valor sea mayor que el tamaño de la clave. Cuando se mueven elementos entre DRAM y SSD, las claves siempre permanecerán en la memoria y solo los valores se moverán al nivel de SSD.

Limitaciones

La organización de datos en niveles tiene las siguientes restricciones:

- Solo puede utilizar la organización de datos en niveles en clústeres que forman parte de un grupo de replicación.
- El tipo de nodo que utilice debe pertenecer a la familia r6gd, que está disponible en las siguientes regiones: `us-east-2`, `us-east-1`, `us-west-2`, `us-west-1`, `eu-west-1`, `eu-central-1`, `eu-north-1`, `eu-west-3`, `ap-northeast-1`, `ap-southeast-1`, `ap-southeast-2`, `ap-south-1`, `ca-central-1` y `sa-east-1`.
- Debe utilizar un motor que sea Valkey 7.2 o una versión posterior, o Redis OSS 6.2 o una versión posterior.
- No se puede restaurar una copia de seguridad de un clúster r6gd en otro clúster a menos que utilice también r6gd.

- No se puede exportar una copia de seguridad a Amazon S3 para clústeres de organización de datos en niveles.
- La migración en línea no se admite para clústeres que se ejecutan en el tipo de nodo r6gd.
- No se admite el escalado desde un clúster de organización de datos en niveles (por ejemplo, un clúster que utiliza un tipo de nodo r6gd) a un clúster que no utiliza la organización de datos en niveles (por ejemplo, un clúster que utiliza un tipo de nodo r6g). Para obtener más información, consulte Escalado de ElastiCache.
- El escalado automático se admite en los clústeres que utilizan la organización de datos en niveles para Valkey 7.2 y versiones posteriores y Redis OSS 7.0.7 y versiones posteriores. Para obtener más información, consulte Clústeres de escalado automático de Valkey y Redis OSS
- La organización de datos en niveles solo admite las políticas maxmemory `volatile-lru`, `allkeys-lru`, `volatile-lfu`, `allkeys-lfu` y `noeviction`.
- Valkey 7.2 y versiones posteriores y Redis OSS 7.0.7 y versiones posteriores admiten el guardado sin ramificaciones. Para obtener más información, consulte Cómo se implementan la sincronización y la copia de seguridad.
- Los artículos de más de 128 MiB no se mueven a SSD.
- A partir de Valkey 8.1 y versiones posteriores, un elemento cuyo tamaño de clave y valor sea inferior a 40 bytes no se moverá al SSD.

Precios

Los nodos R6gd tienen 4,8 veces más capacidad total (memoria + SSD) y pueden ayudarle a ahorrar más del 60 por ciento cuando se ejecuta con la máxima utilización en comparación con los nodos R6g (solo memoria). Para obtener más información, consulte Precios de ElastiCache.

Monitorización

ElastiCache ofrece métricas diseñadas específicamente para supervisar los clústeres de rendimiento que utilizan la organización de datos en niveles. Para supervisar la proporción de elementos en DRAM en comparación con SSD, puede utilizar la métrica `CurrItems` en Métricas de Valkey y Redis OSS. Puede calcular el porcentaje de la siguiente manera: (CurrItems con dimensión: Nivel = Memoria * 100)/(CurrItems sin filtro de dimensión).

Si la política de expulsión configurada lo permite, ElastiCache empezará a expulsar elementos cuando el porcentaje de estos en la memoria disminuya por debajo del 5 %. En los nodos

configurados con una política de no expulsión, en las operaciones de escritura aparecerá un error de memoria insuficiente.

Se sigue recomendando que considere la posibilidad de escalar horizontalmente los clústeres con el modo de clúster habilitado o de escalar verticalmente los clústeres con el modo de clúster deshabilitado cuando el porcentaje de elementos en memoria disminuya por debajo del 5 %. Para obtener más información sobre el escalado, consulte Escalado de clústeres de Valkey o Redis OSS (modo de clúster habilitado). Para obtener más información sobre las métricas para clústeres de Valkey o Redis OSS que utilizan la organización de datos en niveles, consulte Métricas de Valkey y Redis OSS.

Uso de la organización de datos en niveles

Uso de la organización de datos en niveles mediante la Consola de administración de AWS

Al crear un clúster como parte de un grupo de replicación, se utiliza la organización de datos en niveles seleccionando un tipo de nodo de la familia r6gd, como cache.r6gd.xlarge. La selección de ese tipo de nodo habilita automáticamente la organización de datos en niveles.

Para obtener más información sobre la creación de clústeres, consulte Creación de un clúster para Valkey o Redis OSS.

Activación de la organización de datos en niveles mediante la AWS CLI

Al crear un grupo de replicación de utilizando la AWS CLI, puede utilizar la organización de datos en niveles seleccionando un tipo de nodo de la familia r6gd, como cache.r6gd.xlargey configurar el parámetro `--data-tiering-enabled`.

No puede excluirse de la organización de datos en niveles al seleccionar un tipo de nodo de la familia r6gd. Si configura el parámetro `--no-data-tiering-enabled`, la operación no se llevará a cabo correctamente.

Para Linux, macOS o Unix:

```
aws elasticache create-replication-group \
   --replication-group-id redis-dt-cluster \
   --replication-group-description "Redis OSS cluster with data tiering" \
   --num-node-groups 1 \
   --replicas-per-node-group 1 \
   --cache-node-type cache.r6gd.xlarge \
```

```
   --engine redis \
   --cache-subnet-group-name default \
   --automatic-failover-enabled \
   --data-tiering-enabled
```

Para Windows:

```
aws elasticache create-replication-group ^
   --replication-group-id redis-dt-cluster ^
   --replication-group-description "Redis OSS cluster with data tiering" ^
   --num-node-groups 1 ^
   --replicas-per-node-group 1 ^
   --cache-node-type cache.r6gd.xlarge ^
   --engine redis ^
   --cache-subnet-group-name default ^
   --automatic-failover-enabled ^
   --data-tiering-enabled
```

Después de ejecutar esta operación, verá una respuesta parecida a la siguiente:

```
{
    "ReplicationGroup": {
        "ReplicationGroupId": "redis-dt-cluster",
        "Description": "Redis OSS cluster with data tiering",
        "Status": "creating",
        "PendingModifiedValues": {},
        "MemberClusters": [
            "redis-dt-cluster"
        ],
        "AutomaticFailover": "enabled",
        "DataTiering": "enabled",
        "SnapshotRetentionLimit": 0,
        "SnapshotWindow": "06:00-07:00",
        "ClusterEnabled": false,
        "CacheNodeType": "cache.r6gd.xlarge",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

Restauración de datos desde copias de seguridad en clústeres con la organización de datos en niveles

Puede restaurar una copia de seguridad en un nuevo clúster con la organización de datos en niveles habilitada mediante la (Consola), (AWS CLI) o (API de ElastiCache). Cuando crea un clúster mediante tipos de nodos de la familia r6gd, se habilita la organización de datos en niveles.

Restauración de datos desde la copia de seguridad en clústeres con la organización de datos en niveles activada (consola)

Para restaurar una copia de seguridad a un nuevo clúster con la organización de datos en niveles habilitada (consola)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, seleccione Backups (Copias de seguridad).
3. En la lista de copias de seguridad, active la casilla situada a la izquierda del nombre de la copia de seguridad que desea restaurar.
4. Elija Restore (Restaurar).
5. Complete el cuadro de diálogo Restore Cluster. Asegúrese de completar todos los campos Required (Obligatorios) y los demás campos cuyos valores predeterminados desee modificar.
    1. ID del clúster: obligatorio. Se trata del nombre del nuevo clúster.
    2. Modo de clúster habilitado (escalado horizontal): elija esta opción para clústeres de Valkey o Redis OSS (modo de clúster habilitado).
    3. Tipo de nodo: especificar cache.r6gd.xlarge o cualquier otro tipo de nodo de la familia r6gd.
    4. Number of Shards: elija el número de fragmentos que desea en el nuevo clúster (API/CLI: grupos de nodos).
    5. Replicas per Shard: elija el número de nodos de réplica de lectura que desea en cada fragmento.
    6. Ranuras y espacios de claves: elija cómo desea que se distribuyan las claves entre las particiones. Si decide especificar las distribuciones de claves, complete la tabla especificando los rangos de claves para cada fragmento.
    7. Availability zone(s): especifique cómo desea que se seleccionen las zonas de disponibilidad del clúster.
    8. Port: modifique este valor solo si desea que el nuevo clúster use un puerto distinto.

9. Choose a VPC: elija la VPC en la que va a crear este clúster.
10. Grupo de parámetros: elija un grupo de parámetros que reserve memoria suficiente para la sobrecarga de Valkey o Redis OSS del tipo de nodo que haya seleccionado.

6. Cuando esté conforme con los ajustes, elija Crear.

Para obtener más información sobre la creación de clústeres, consulte Creación de un clúster para Valkey o Redis OSS.

Restauración de datos desde copias de seguridad en clústeres con la organización de datos en niveles habilitada (AWS CLI)

Al crear un grupo de replicación de utilizando la AWS CLI, la organización de datos en niveles se utiliza de forma predeterminada seleccionando un tipo de nodo de la familia r6gd, como cache.r6gd.xlarge y configurar el parámetro `--data-tiering-enabled`.

No puede excluirse la organización de datos en niveles al seleccionar un tipo de nodo de la familia r6gd. Si configura el parámetro `--no-data-tiering-enabled`, la operación no se llevará a cabo correctamente.

Para Linux, macOS o Unix:

```
aws elasticache create-replication-group \
   --replication-group-id redis-dt-cluster \
   --replication-group-description "Redis OSS cluster with data tiering" \
   --num-node-groups 1 \
   --replicas-per-node-group 1 \
   --cache-node-type cache.r6gd.xlarge \
   --engine redis \
   --cache-subnet-group-name default \
   --automatic-failover-enabled \
   --data-tiering-enabled \
   --snapshot-name my-snapshot
```

Para Linux, macOS o Unix:

```
aws elasticache create-replication-group ^
   --replication-group-id redis-dt-cluster ^
   --replication-group-description "Redis OSS cluster with data tiering" ^
   --num-node-groups 1 ^
   --replicas-per-node-group 1 ^
```

```
    --cache-node-type cache.r6gd.xlarge ^
    --engine redis ^
    --cache-subnet-group-name default ^
    --automatic-failover-enabled ^
    --data-tiering-enabled ^
    --snapshot-name my-snapshot
```

Después de ejecutar esta operación, verá una respuesta parecida a la siguiente:

```
{
    "ReplicationGroup": {
        "ReplicationGroupId": "redis-dt-cluster",
        "Description": "Redis OSS cluster with data tiering",
        "Status": "creating",
        "PendingModifiedValues": {},
        "MemberClusters": [
            "redis-dt-cluster"
        ],
        "AutomaticFailover": "enabled",
        "DataTiering": "enabled",
        "SnapshotRetentionLimit": 0,
        "SnapshotWindow": "06:00-07:00",
        "ClusterEnabled": false,
        "CacheNodeType": "cache.r6gd.xlarge",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

Preparación de un clúster en ElastiCache

A continuación, puede encontrar instrucciones sobre cómo crear un clúster a través de la consola de ElastiCache, la AWS CLI o la API de ElastiCache.

También puede crear un clúster de ElastiCache con AWS CloudFormation. Para obtener más información, consulte AWS::ElastiCache::CacheCluster en la Guía del usuario de AWS CloudFormation, que contiene directrices acerca de cómo implementar este enfoque.

Siempre que cree un clúster o un grupo de reproducción, es conveniente realizar algunos preparativos para que no sea necesario actualizar o efectuar cambios de inmediato.

Temas

- Determinación de los requisitos del clúster de ElastiCache
- Elección del tamaño del nodo

Determinación de los requisitos del clúster de ElastiCache

Preparación

Conocer las respuestas a las siguientes preguntas ayuda a que la creación del clúster de ElastiCache sea más fluida:

- ¿Qué tipo de instancia de nodo necesita?

  Para obtener instrucciones sobre cómo elegir un tipo de nodo de instancia, consulte Elección del tamaño del nodo.
- ¿Lanzará su clúster en una nube virtual privada (VPC) en función de Amazon VPC?

  > **Important**
  >
  > Si va a lanzar su clúster en una VPC, debe asegurarse de crear un grupo de subredes en la misma VPC antes de comenzar a crear un clúster. Para obtener más información, consulte Subredes y grupos de subredes.
  > ElastiCache se ha diseñado para el acceso desde AWS mediante Amazon EC2. Sin embargo, si se lanza en una VPC basada en Amazon VPC y el clúster se encuentra en una VPC, se puede proporcionar acceso desde fuera de AWS. Para obtener más información, consulte Acceso a recursos de ElastiCache desde fuera de AWS.
- ¿Necesita personalizar los valores de algún parámetro?

  Si lo hace, cree un grupo de parámetros personalizado. Para obtener más información, consulte Creación de un grupo de parámetros de ElastiCache.

  Si ejecuta Valkey o Redis OSS, considere establecer `reserved-memory` o `reserved-memory-percent`. Para obtener más información, consulte Administración de la memoria reservada para Valkey y Redis OSS.
- ¿Necesita crear su propio grupo de seguridad de VPC?

  Para obtener más información, consulte Seguridad en la VPC.
- ¿Cómo pretende implementar la tolerancia a errores?

Para obtener más información, consulte Mitigación de errores.

Temas

- Requisitos de procesador y memoria de ElastiCache
- Configuración de los clústeres de Memcached
- Configuración de clústeres de Valkey y Redis OSS
- Requisitos de escalado de ElastiCache
- Requisitos de acceso a ElastiCache
- Requisitos de región, zona de disponibilidad y zona local para ElastiCache

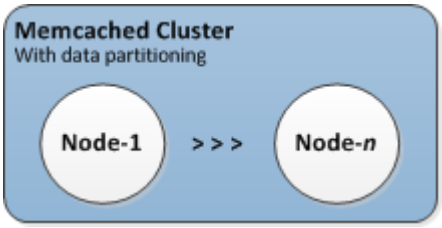
Requisitos de procesador y memoria de ElastiCache

El componente básico de Amazon ElastiCache es el nodo. Los nodos se configuran individualmente o en agrupaciones para formar clústeres. A la hora de determinar el tipo de nodo que desea utilizar para el clúster, tenga en cuenta la configuración del nodo del clúster y la cantidad de datos que tiene que almacenar.

El motor Memcached tiene varios subprocesos, por lo que el número de núcleos de un nodo repercute en la potencia de cómputo disponible para el clúster.

Configuración de los clústeres de Memcached

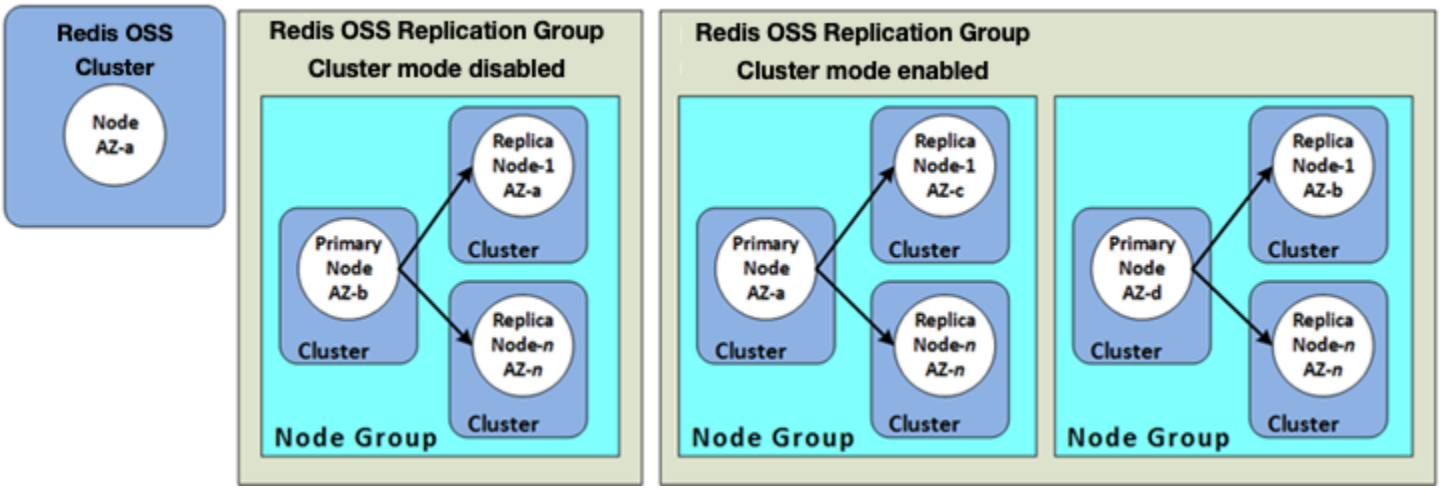
Los clústeres de ElastiCache for Memcached se componen de 1 a 60 nodos. Los datos de un clúster de Memcached están particionados entre los nodos del clúster. La aplicación se conecta con un clúster de Memcached mediante una dirección de red denominada punto de enlace. Cada nodo de un clúster de Memcached tiene su propio punto de enlace. La aplicación utiliza este punto de enlace para leer o escribir en dicho nodo. Además de los puntos de enlace del nodo, el clúster de Memcached tiene un punto de enlace denominado punto de enlace de configuración. Su aplicación puede usar este punto de conexión para leer o escribir en el clúster; de este modo, es posible dejar que la detección automática determine en qué nodo leer o escribir.



Para obtener más información, consulte Administración de clústeres en ElastiCache.

Configuración de clústeres de Valkey y Redis OSS

Los clústeres de ElastiCache para Valkey y Redis OSS constan de 0 a 500 particiones (también denominadas grupos de nodos). Los datos de un clúster de Valkey o Redis OSS se dividen entre las particiones del clúster. La aplicación se conecta con un clúster de Valkey o Redis OSS a través una dirección de red denominada punto de conexión. Los nodos de una partición de Valkey o Redis OSS adoptan uno de estos dos roles: uno adopta el rol de nodo principal de lectura/escritura y los otros adoptan el rol de nodos secundarios de solo lectura (llamados también réplicas de lectura). Además de los puntos de conexión del nodo, el clúster de Valkey o Redis OSS tiene un punto de conexión denominado punto de conexión de configuración. Su aplicación puede utilizar este punto de conexión para leer o escribir en el clúster y no tener que determinar de qué nodo efectuar la lectura o hasta cuál escribir en ElastiCache para Redis OSS.



Para obtener más información, consulte Administración de clústeres en ElastiCache.

Requisitos de escalado de ElastiCache

Todos los clústeres pueden ampliarse mediante la creación de un nuevo clúster con el nuevo tipo de nodo, que es más grande. Al escalar verticalmente un clúster de Memcached, el nuevo clúster comenzará vacío. Al escalar verticalmente un clúster de Valkey o Redis OSS, puede inicializarlo desde una copia de seguridad para evitar que el nuevo clúster esté vacío.

Los clústeres de Amazon ElastiCache for Memcached pueden reducirse o escalarse horizontalmente. Para escalar un clúster de Memcached de forma ascendente o descendente, simplemente agregue o quite nodos del clúster. Si ha habilitado la detección automática y su aplicación se conecta al punto de enlace de configuración del clúster, no es necesario que realice ningún cambio en la aplicación al agregar o quitar nodos.

Para obtener más información, consulte la sección Escalado de ElastiCache de esta guía.

Requisitos de acceso a ElastiCache

Por diseño, el acceso a los clústeres de Amazon ElastiCache se realiza desde instancias de Amazon EC2. El acceso de red a un clúster de ElastiCache se encuentra limitado a la cuenta que ha creado el clúster. Por lo tanto, antes de poder acceder a un clúster desde una instancia de Amazon EC2, debe autorizar a dicha instancia para el acceso al clúster. Los pasos para ello varían en función de si realiza el lanzamiento en EC2-VPC o en EC2-Classic.

Si ha lanzado el clúster en EC2-VPC, necesita conceder acceso de red al clúster. Si ha lanzado un clúster en EC2-Classic, debe conceder al grupo de seguridad de Amazon Elastic Compute Cloud (EC2) asociado a la instancia acceso a su grupo de seguridad de ElastiCache. Para obtener instrucciones detalladas, consulte Paso 3. Autorización del acceso al clúster en esta guía.

Requisitos de región, zona de disponibilidad y zona local para ElastiCache

Amazon ElastiCache admite todos las regiones de AWS. Al ubicar sus clústeres de ElastiCache en una región de AWS cercana a su aplicación, puede reducir la latencia. Si el clúster tiene varios nodos, ubicar los nodos en distintas zonas de disponibilidad o en Local Zones puede reducir el impacto de los errores en el clúster.

Para obtener más información, consulte los siguientes temas:

- Elección de regiones y zonas de disponibilidad para ElastiCache
- Uso de Local Zones con ElastiCache
- Mitigación de errores

Elección del tamaño del nodo

El tamaño del nodo que seleccione para el clúster de ElastiCache afectará a los costos, al rendimiento y a la tolerancia a errores.

Tamaño del nodo (Valkey y Redis OSS)

Para obtener información sobre las ventajas de los procesadores Graviton, consulte Procesador AWS Graviton.

Responder las siguientes preguntas puede ayudar a determinar el tipo de nodo mínimo que necesita para su implementación de Valkey o Redis OSS:

- ¿Espera cargas de trabajo limitadas por el rendimiento con múltiples conexiones de cliente?

  Si este es el caso y está ejecutando la versión 5.0.6 o posteriores de Redis OSS, puede obtener un mejor rendimiento y latencia con nuestra característica de E/S mejorada, donde las CPU disponibles se utilizan para descargar las conexiones del cliente en nombre del motor de Redis OSS. Si ejecuta la versión 7.0.4 o posteriores de Redis OSS, además de la E/S mejorada, obtendrá una aceleración adicional con la multiplexación de E/S mejorada, en la que cada subproceso de E/S de red específico canaliza los comandos desde varios clientes al motor de Redis OSS, aprovechando así la capacidad de Redis OSS de procesar comandos en lotes de manera eficiente. En ElastiCache para Redis 7.1 y versiones posteriores, hemos ampliado la funcionalidad mejorada de los subprocesos de E/S para gestionar también la lógica de la capa de presentación. Al hablar de la capa de presentación, nos referimos a los subprocesos de E/S mejorados. Ahora, no solo leen la entrada del cliente, sino que también analizan la entrada en el formato de comandos binarios de Redis OSS; luego, se envía al subproceso principal para la ejecución, lo que aumenta el rendimiento. Consulte la entrada de blog y la página de versiones compatibles para obtener más información.

- ¿Tiene cargas de trabajo que acceden regularmente a un pequeño porcentaje de sus datos?

  Si este es el caso y está ejecutando en el motor de Redis OSS versión 6.2 o posteriores, puede aprovechar la organización de datos en niveles eligiendo el tipo de nodo r6gd. Con la organización de datos en niveles, los datos menos usados recientemente se almacenan en SSD. Cuando se recupera, hay un pequeño costo de latencia, que se equilibra con el ahorro de costos. Para obtener más información, consulte Organización de datos en niveles en ElastiCache.

  Para obtener más información, consulte Tipos de nodos compatibles.

- ¿Cuánta memoria total necesita para sus datos?

  Para obtener una estimación general, tome el tamaño de los elementos que desea almacenar en la caché. Multiplique este tamaño por el número de elementos que desea conservar en la caché al mismo tiempo. Para obtener una estimación razonable del tamaño de los elementos, serialice los elementos de la caché y cuente los caracteres. A continuación, divida esto entre el número de particiones de su clúster.

  Para obtener más información, consulte Tipos de nodos compatibles.

- ¿Qué versión de Redis OSS está ejecutando?

  Las versiones de Redis OSS anteriores a la 2.8.22 requieren que reserve más memoria para la conmutación por error, la sincronización de instantáneas y la promoción de réplicas a operaciones

principales. Este requisito se debe a que debe disponer de suficiente memoria disponible para todas las operaciones de escritura que se producen durante el proceso.

La versión 2.8.22 de Redis OSS y las versiones posteriores usan un proceso de guardado sin ramificaciones que requiere menos memoria disponible que el proceso anterior.

Para obtener más información, consulte los siguientes temas:

- Cómo se implementan la sincronización y la copia de seguridad
- Forma de garantizar que dispone de memoria suficiente para crear una instantánea de Valkey o Redis OSS

- ¿Qué volumen de operaciones de escritura realiza su aplicación?

  Las aplicaciones que realizan un elevado volumen de operaciones de escritura pueden requerir un mayor volumen de memoria disponible (memoria no usada por los datos) a la hora de tomar instantáneas o en casos de conmutación por error. Cuando se aplica el proceso `BGSAVE`, debe tener suficiente memoria que no sea utilizada por los datos para alojar todas las escrituras que se producen durante el proceso `BGSAVE`. Algunos ejemplos son: tomar una instantánea, sincronizar un clúster principal con una réplica en un clúster y habilitar la característica de archivo de solo anexado (AOF). Otro ejemplo es cuando se promueve una réplica a principal (si tiene habilitado Multi-AZ). El peor de los casos es cuando todos los datos se reescriben durante el proceso. En este caso, necesita un tamaño de instancia de nodo con el doble de la memoria necesaria solo para los datos.

  Para obtener información más detallada, consulte Forma de garantizar que dispone de memoria suficiente para crear una instantánea de Valkey o Redis OSS.

- ¿Su implementación será un clúster de Valkey o Redis OSS independiente (modo de clúster deshabilitado) o un clúster de Valkey o Redis OSS (modo de clúster habilitado) con varias particiones?

  Clúster de Valkey o Redis OSS (modo de clúster deshabilitado)

  Si va a implementar un clúster de Valkey o Redis OSS (modo de clúster deshabilitado), su tipo de nodo debe tener capacidad para todos los datos más una capacidad adicional, tal como se describe en el punto anterior.

  Por ejemplo, supongamos que estima que el tamaño total de todos los elementos es de 12 GB. En este caso, puede utilizar un nodo de `cache.m3.xlarge` con 13.3 GB de memoria o un nodo de `cache.r3.large` con 13.5 GB de memoria. Sin embargo, es posible que necesite más

memoria para las operaciones BGSAVE. Si la aplicación tiene un gran volumen de operaciones de escritura, duplique los requisitos de memoria a 24 GB como mínimo. Por lo tanto, utilice un `cache.m3.2xlarge` con 27.9 GB de memoria o un `cache.r3.xlarge` con 30.5 GB de memoria.

Valkey o Redis OSS (modo de clúster habilitado) con varias particiones

Si va a implementar un clúster de Valkey o Redis OSS (modo de clúster habilitado) con varias particiones, el tipo de nodo debe tener capacidad para `bytes-for-data-and-overhead / number-of-shards` bytes de datos.

Por ejemplo, suponga que estima que el tamaño total de todos los elementos es de 12 GB y tiene dos particiones. En este caso, puede utilizar un nodo de `cache.m3.large` con 6.05 GB de memoria (12 GB/2). Sin embargo, es posible que necesite más memoria para las operaciones BGSAVE. Si la aplicación tiene un gran volumen de operaciones de escritura, duplique los requisitos de memoria a 12 GB por partición como mínimo. Por lo tanto, utilice un `cache.m3.xlarge` con 13.3 GB de memoria o un `cache.r3.large` con 13.5 GB de memoria.

- ¿Utiliza Local Zones?

  Local Zones permite colocar recursos como un clúster de ElastiCache en varias ubicaciones cercanas a los usuarios. Sin embargo, cuando elija el tamaño del nodo, tenga en cuenta que, independientemente de los requisitos de capacidad, los tamaños de nodo disponibles tienen los siguientes límites en este momento:

  - Generación actual:

    Tipos de nodos M: `cache.m5.large`, `cache.m5.xlarge`, `cache.m5.2xlarge`, `cache.m5.4xlarge`, `cache.m5.12xlarge`, `cache.m5.24xlarge`

    Tipos de nodos R: `cache.r5.large`, `cache.r5.xlarge`, `cache.r5.2xlarge`, `cache.r5.4xlarge`, `cache.r5.12xlarge`, `cache.r5.24xlarge`

    Tipos de nodos T: `cache.t3.micro`, `cache.t3.small`, `cache.t3.medium`

Mientras el clúster se está ejecutando, podrá monitorear el uso de memoria, el uso del procesador, las métricas de los errores de caché y los aciertos de caché que se publican en CloudWatch. Es posible que observe que el clúster no tiene la tasa de aciertos que desea o que las claves se expulsan con demasiada frecuencia. En estos casos, puede elegir un tamaño de nodo diferente con mayores especificaciones de memoria y CPU.

Cuando supervise el uso de la CPU, recuerde que Valkey y Redis OSS tienen un subproceso único. Por lo tanto, multiplique el uso de la CPU por el número de núcleos de CPU para obtener el uso real. Por ejemplo, una CPU de cuatro núcleos cuya tasa de uso es del 20 % indica que el núcleo de Redis OSS se encuentra en uso al 80 %.

Tamaño del nodo (Memcached)

Los clústeres de Memcached contienen uno o varios nodos entre los que se particionan los datos del clúster. Por ello, las necesidades de memoria del clúster y la memoria de un nodo están relacionadas, pero no son la misma cosa. Puede alcanzar la capacidad de memoria del clúster necesaria con varios nodos de gran tamaño o varios nodos más pequeños. Además, a medida que cambien sus necesidades, puede agregar nodos al clúster o eliminarlos y, por lo tanto, pagar solo por aquello que necesite.

La capacidad total de memoria de su clúster se calcula multiplicando el número de nodos del clúster por la capacidad de RAM de cada nodo, tras haberle restado la carga general del sistema. La capacidad de cada nodo depende del tipo de nodo.

```
cluster_capacity = number_of_nodes * (node_capacity - system_overhead)
```

El número de nodos del clúster es un factor clave para la disponibilidad de su clúster con Memcached. El error de un único nodo puede repercutir en la disponibilidad de su aplicación y en la carga de la base de datos de backend. En tal caso, ElastiCache aprovisiona un nodo de reemplazo para el nodo con error y propaga los datos. Para reducir este impacto en la disponibilidad, distribuya su memoria y su capacidad informática en un mayor número de nodos, cada uno con menos capacidad, en lugar de usar menos nodos de mayor capacidad.

En un escenario en el que desea disponer de 35 GB de memoria caché, puede realizar cualquiera de las siguientes configuraciones:

- 11 `cache.t2.medium` nodos con 3,22 GB de memoria y 2 subprocesos en cada uno = 35,42 GB y 22 subprocesos.
- 6 `cache.m4.large` nodos con 6,42 GB de memoria y 2 subprocesos en cada uno = 38,52 GB y 12 subprocesos.
- 3 `cache.r4.large` nodos con 12,3 GB de memoria y 2 subprocesos en cada uno = 36,90 GB y 6 subprocesos.
- 3 `cache.m4.xlarge` nodos con 14,28 GB de memoria y 4 subprocesos en cada uno = 42,84 GB y 12 subprocesos.

Comparación de opciones de nodos

Tipo de nodo	Memoria (en GiB)	Núcleos	Costo por horas*	Nodos necesarios	Memoria total (en GiB)	Núcleos totales	Costo mensual
cache.t2.medium	3,22	2	0,068 USD	11	35,42	22	538,56 USD
cache.m4.large	6,42	2	0,156 USD	6	38,52	12	673,92 USD
cache.m4.xlarge	14,28	4	0,311 USD	3	42,84	12	671,76 USD
cache.m5.xlarge	12,93	4	0,311 USD	3	38,81	12	671,76 USD
cache.m6g.large	6,85	2	0,147 USD	6	41,1	12	635 USD
cache.r4.large	12.3	2	0,228 USD	3	36,9	6	492,48 USD
cache.r5.large	13,07	2	0,216 USD	3	39,22	6	466,56 USD
cache.r6g.large	13,07	2	0,205 USD	3	42,12	6	442 USD

* Costo por hora por nodo al 8 de octubre de 2020.

Costo mensual con un 100 % de uso durante 30 días (720 horas).

Estas opciones proporcionan una capacidad de memoria similar pero con diferencias de costo y capacidad de cómputo. Para comparar los costos de sus opciones específicas, consulte Precios de Amazon ElastiCache.

Para clústeres que ejecutan Memcached, parte de la memoria disponible en cada nodo se usa para la conexión. Para obtener más información, consulte Capacidad adicional para conexiones de Memcached.

El uso de varios nodos requiere distribuir las claves entre ellos. Cada nodo tiene su propio punto de conexión. Para facilitar la administración de los puntos de enlace, utilice la característica de detección automática de ElastiCache, que permite a los programas de cliente identificar de forma automática todos los nodos de un clúster. Para obtener más información, consulte Identificación automática de los nodos en el clúster (Memcached).

En algunos casos, es posible que no se encuentre seguro de cuánta capacidad necesita. Si es así, para las pruebas recomendamos comenzar con un nodo `cache.m5.large`. A continuación, monitoree el uso de memoria, el uso de la CPU y la tasa de aciertos de caché con las métricas de ElastiCache que se publican en Amazon CloudWatch. A fin de obtener más información sobre las métricas de CloudWatch para ElastiCache, consulte Monitoreo del uso con métricas de CloudWatch. Para las cargas de trabajo de mayor tamaño y de producción, los nodos R5 ofrecen el mejor rendimiento y el precio de RAM más ajustado.

Si su clúster no tiene la tasa deseada, podrá agregar más nodos fácilmente para aumentar la memoria total disponible en el clúster.

Si el clúster se encuentra limitado por la CPU pero tiene una tasa suficiente, configure un clúster nuevo con un tipo de nodo que ofrezca mayor potencia informática.

Creación de un clúster para Valkey o Redis OSS

En los siguientes ejemplos, se muestra cómo crear un clúster de Valkey o Redis OSS con la Consola de administración de AWS, la AWS CLI y la API de ElastiCache.

Creación un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) (consola)

ElastiCache admite la replicación si se utiliza el motor de Valkey o Redis OSS. Para supervisar la latencia entre cuando se escriben los datos en un clúster principal de lectura/escritura de Valkey o Redis OSS y cuando se propagan a un clúster secundario de solo lectura, ElastiCache añade al clúster una clave especial, `ElastiCacheMasterReplicationTimestamp`. Esta clave es la hora de tiempo universal (UCT). Puesto que se puede añadir un clúster de Valkey o Redis OSS más adelante a un grupo de replicación, esta clave se incluye en todos los clústeres de Valkey o Redis OSS, incluso si inicialmente no son miembros de un grupo de replicación. Para obtener más información acerca de los grupos de reproducción, consulte Alta disponibilidad a través de grupos de reproducción.

Para crear un clúster de Valkey o Redis OSS (modo clúster deshabilitado), siga los pasos de Creación de un clúster de Valkey (modo de clúster deshabilitado) (consola).

En cuanto el estado de su clúster sea available (disponible), podrá conceder a Amazon EC2 acceso a este, así como conectarse a él y comenzar a utilizarlo. Para obtener más información, consulte Paso 3. Autorización del acceso al clúster y Paso 4. Conexión al nodo de un clúster.

> **Important**
>
> Cuando su clúster esté disponible, se cobrará por cada hora u hora parcial que el clúster esté activo, incluso si no lo está utilizando de forma activa. Para dejar de incurrir en cargos por este clúster, debe eliminarlo. Consulte Eliminación de un clúster en ElastiCache.

Creación de un clúster de Valkey o Redis OSS (modo de clúster habilitado) (consola)

Si ejecuta Redis OSS 3.2.4 o una versión posterior, puede crear un clúster de Valkey o Redis OSS (modo de clúster habilitado). Los clústeres de Valkey o Redis OSS (modo de clúster habilitado) permiten dividir los datos entre 1 y 500 particiones (API/CLI: grupos de nodos), pero con algunas limitaciones. Para obtener una comparación de Valkey o Redis OSS (modo de clúster deshabilitado) y Valkey o Redis OSS (modo de clúster habilitado), consulte Motores y versiones compatibles.

Creación de un clúster de Valkey o Redis OSS (modo de clúster habilitado) con la consola de ElastiCache

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En la lista de la esquina superior derecha, elija la región de AWS en la que desea lanzar este clúster.
3. En el panel de navegación, seleccione Get started (Introducción).
4. Elija Create VPC (Crear una VPC) y siga los pasos que se detallan en Creating a Virtual Private Cloud (VPC) (Creación de una nube privada virtual).
5. En la página del panel de ElastiCache, elija Crear clúster y luego elija Crear clúster de Valkey o Crear clúster de Redis OSS.
6. En Cluster settings (Configuración de clúster), realice la siguiente operación:
   a. Elija Configure and create a new cluster (Configurar y crear un nuevo clúster).
   b. Para Cluster mode (Modo clúster), elija Enabled (Habilitado).
   c. Para Cluster info (Información del clúster) introduzca un valor para Name (Nombre).
   d. (Opcional) Introduzca un valor para Description (Descripción).
7. En Location (Ubicación):

   AWS Cloud

   1. Para Nube de AWS, le recomendamos que acepte la configuración predeterminada de Multi-AZ y Conmutación por error automática. Para obtener más información, consulte Minimización del tiempo de inactividad en ElastiCache para Redis OSS con Multi-AZ.
   2. En Cluster settings (Configuración del clúster)
      a. Para Engine version (Versión del motor), elija una versión disponible.
      b. En Port (Puerto), utilice el puerto predeterminado, 6379. Si tiene un motivo para utilizar un puerto diferente, escriba el número de puerto.
      c. En Grupo de parámetros, elija un grupo de parámetros o cree uno nuevo. Los grupos de parámetros controlan los parámetros de tiempo de ejecución de su clúster. Para obtener más información acerca de los grupos de parámetros, consulte Parámetros de Valkey y Redis OSS y Creación de un grupo de parámetros de ElastiCache.

> **Note**
>
> Cuando se selecciona un grupo de parámetros para definir los valores de configuración del motor, dicho grupo de parámetros se aplica a todos los clústeres del almacén de datos global. En la página Parameter Groups (Grupos de parámetros), el atributo Global sí/no, indica si un grupo de parámetros forma parte de un almacén de datos global.

d. Para Node type, elija la flecha hacia abajo (▼).
En el cuadro de diálogo Cambiar tipo de nodo, elija un valor para Familia de instancias para el tipo de nodo que desea. A continuación, elija el tipo de nodo que desea utilizar para este clúster y, a continuación, elija Guardar.

   Para obtener más información, consulte Elección del tamaño del nodo.

   Si elige un tipo de nodo r6gd, la organización de datos en niveles se activará automáticamente. Para obtener más información, consulte Organización de datos en niveles en ElastiCache.

e. En Número de particiones, elija el número de particiones (particiones/grupos de nodos) que desea para este clúster de Valkey o Redis OSS (modo de clúster habilitado).

   En algunas versiones de Valkey o Redis OSS (modo de clúster habilitado), puede cambiar dinámicamente el número de particiones del clúster:

   - Redis OSS 3.2.10 y versiones posteriores: si el clúster ejecuta Redis OSS 3.2.10 o versiones posteriores, puede cambiar el número de particiones del clúster de forma dinámica. Para obtener más información, consulte Escalado de clústeres de Valkey o Redis OSS (modo de clúster habilitado).
   - Otras versiones de Redis OSS: si su clúster ejecuta una versión de Redis OSS anterior a 3.2.10, existe otro enfoque. Para cambiar el número de particiones del clúster en este caso, cree un nuevo clúster con el nuevo número de particiones. Para obtener más información, consulte Restauración desde una copia de seguridad a una nueva caché.

f. En Réplicas por partición, elija el número de nodos de réplica de lectura que desea en cada partición.

Existen las siguientes restricciones para Valkey o Redis OSS (modo de clúster habilitado).

- Si tiene habilitado Multi-AZ, asegúrese de tener al menos una réplica por partición.
- El número de réplicas es el mismo para cada fragmento al crear el clúster utilizando la consola.
- El número de réplicas de lectura por fragmento es fijo y no puede modificarse. Si se da cuenta de que necesita más o menos réplicas por fragmento (API/CLI: grupo de nodos), debe crear un nuevo clúster con el nuevo número de réplicas. Para obtener más información, consulte Tutorial: propagación de un nuevo clúster basado en nodos con una copia de seguridad creada externamente.

3. En Connectivity (Conectividad)

   a. Para el Network type (Tipo de red), elija las versiones IP que admitirá este clúster.

   b. En Subnet groups (Grupos de subredes), elija la subred que desea aplicar a este clúster. ElastiCache utiliza dicho grupo de subredes para elegir una subred y direcciones IP pertenecientes a dicha subred a fin de asociarlas a sus nodos. Los clústeres de ElastiCache requieren una subred de doble pila con direcciones IPv4 e IPv6 asignadas para funcionar en modo de doble pila y una subred solo IPv6 para funcionar como solo IPv6.

      Al crear un nuevo grupo de subredes, escriba el VPC ID (ID de VPC) al que pertenece.

      Seleccione un Discovery IP type (Tipo de IP de detección). Solo se devuelven las direcciones IP del protocolo elegido.

      Para obtener más información, consulte:

      - Elección de un tipo de red en ElastiCache.
      - Cree una subred en la VPC.

      Si es Uso de Local Zones con ElastiCache, debe crear o elegir una subred que se encuentre en la zona local.

      Para obtener más información, consulte Subredes y grupos de subredes.

4. En Availability zone placements Ubicaciones de las zonas de disponibilidad), tiene dos opciones:

   - No preference (Sin preferencia): ElastiCache elige la zona de disponibilidad.
   - Specify availability zones (Especificar zonas de disponibilidad): especifique la zona de disponibilidad para cada clúster.

     Si decide especificar las zonas de disponibilidad, para cada clúster en cada fragmento, elija la zona de disponibilidad de la lista.

   Para obtener más información, consulte Elección de regiones y zonas de disponibilidad para ElastiCache.

5. Elija Next (Siguiente).
6. En Configuración avanzada de Valkey o Configuración avanzada de Redis OSS o

   - Para Security (Seguridad):

     i. Para cifrar sus datos, tiene las siguientes opciones:

     - Encryption at rest (Cifrado en reposo): permite el cifrado de los datos almacenados en el disco. Para obtener más información, consulte Cifrado en reposo.

       > **Note**
       >
       > Tiene la opción de suministrar una clave de cifrado distinta al elegir Customer Managed AWS KMS key (Clave KMS de administrada por el cliente) y elegir la clave. Para obtener más información, consulte Uso de claves de AWS KMS administradas por el cliente.

     - Encryption in-transit (Cifrado en tránsito): permite el cifrado de datos del cable. Para obtener más información, consulte Cifrado en tránsito. Para Valkey 7.2 y versiones posteriores o Redis OSS 6.0 y posteriores, si habilita el cifrado en tránsito, se le pedirá que especifique una de las siguientes opciones de Control de acceso:

- No Access Control (Sin control de acceso): este es el valor predeterminado. Esto indica que no hay restricciones en el acceso de los usuarios al clúster.
- User Group Access Control List (Lista de control de acceso del grupo de usuarios): seleccione un grupo de usuarios con un conjunto definido de usuarios que pueden acceder al clúster. Para obtener más información, consulte Administración de grupos de usuarios con la consola y la CLI.
- Usuario predeterminado con el comando AUTH: un mecanismo de autenticación para el servidor de Valkey o Redis OSS. Para obtener más información, consulte AUTH.

- AUTH: un mecanismo de autenticación para el servidor de Valkey o Redis OSS. Para obtener más información, consulte AUTH.

> **Note**
>
> Para las versiones de Redis OSS a partir de la 3.2.6, excepto la versión 3.2.10, AUTH es la única opción.

ii. En Grupos de seguridad, elija los grupos de seguridad que desea para este clúster. Un grupo de seguridad actúa como un firewall para controlar el acceso de red al clúster. Puede utilizar el grupo de seguridad predeterminado para la VPC o crear uno nuevo.

Para obtener más información sobre los grupos de seguridad, consulte Grupos de seguridad de su VPC en la Guía del usuario de Amazon VPC.

7. En las copias de seguridad automáticas programadas regularmente, seleccione Enable automatic backups (Habilitar copias de seguridad automáticas) y, a continuación, escriba el número de días que desea conservar cada copia de seguridad automática antes de que se elimine automáticamente. Si no quiere tener copias de seguridad automáticas programadas regularmente, desactive la casilla de verificación Enable automatic backups (Habilitar copias de seguridad automáticas). En cualquier caso, siempre tendrá la opción de crear copias de seguridad manuales.

   Para obtener más información acerca de la copia de seguridad y restauración, consulte Instantánea y restauración.

8. (Opcional) Especifique un periodo de mantenimiento. El periodo de mantenimiento hace alusión al tiempo, generalmente una hora, de cada semana en la que ElastiCache programa el mantenimiento del sistema en su clúster. Puede permitir que ElastiCache elija el día y la hora de su periodo de mantenimiento (No preference [Sin preferencia]), o bien puede elegir el día, la hora y la duración por su cuenta (Specify maintenance window [Especificar periodo de mantenimiento]). Si elige Especificar el periodo de mantenimiento en las listas, seleccione Día de inicio, Hora de inicio y Duración (en horas) para el periodo de mantenimiento. Todas las horas se indican en UCT.

   Para obtener más información, consulte Administración del mantenimiento de clústeres de ElastiCache.
9. (Opcional) Para Logs (Registros):

   - En Log format (Formato de registro), elija Text o JSON.
   - En Destination Type (Tipo de destino), elija CloudWatch Logs o Kinesis Firehose.
   - En Destino de registro, elija Crear nuevo e introduzca el nombre del grupo de registro de Registros de CloudWatch o el nombre del flujo de Firehose, o elija Seleccionar existente y, a continuación, elija el nombre del grupo de registro de Registros de CloudWatch o el nombre del flujo de Firehose.
10. Para Tags (Etiquetas), para ayudarlo a administrar sus clústeres y otros recursos de ElastiCache, puede asignar sus propios metadatos a cada recurso en forma de etiquetas. Para obtener más información, consulte Etiquetado de los recursos de ElastiCache.
11. Elija Siguiente.
12. Revise todas las entradas y opciones y, a continuación, realice todos los cambios necesarios. Cuando haya terminado, elija Create (Crear).

On premises

1. Para On premises (En las instalaciones), le recomendamos que deje Auto-failover (Conmutación por error automática) habilitada. Para obtener más información, consulte Minimización del tiempo de inactividad en ElastiCache para Redis OSS con Multi-AZ.
2. Siga los pasos de Uso de Outposts.

Para crear el equivalente mediante la AWS CLI o la API de ElastiCache en lugar de la consola de ElastiCache, consulte lo siguiente:

- API: CreateReplicationGroup
- CLI: create-replication-group

En cuanto el estado de su clúster sea available, podrá conceder a EC2 acceso a este, así como conectarse a él y comenzar a utilizarlo. Para obtener más información, consulte Paso 3. Autorización del acceso al clúster y Paso 4. Conexión al nodo de un clúster.

> **Important**
>
> Cuando su clúster esté disponible, se cobrará por cada hora u hora parcial que el clúster esté activo, incluso si no lo está utilizando de forma activa. Para dejar de incurrir en cargos por este clúster, debe eliminarlo. Consulte Eliminación de un clúster en ElastiCache.

Creación de un clúster (AWS CLI)

Para crear un clúster mediante la AWS CLI, utilice el comando `create-cache-cluster`.

> **Important**
>
> Cuando su clúster esté disponible, se cobrará por cada hora u hora parcial que el clúster esté activo, incluso si no lo está utilizando de forma activa. Para dejar de incurrir en cargos por este clúster, debe eliminarlo. Consulte Eliminación de un clúster en ElastiCache.

Creación de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) (CLI)

Example – Un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) sin réplicas de lectura

El siguiente código de la CLI crea un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) sin réplicas.

> **Note**
>
> Al crear clúster utilizando un tipo de nodo de la familia r6gd, debe pasar el parámetro `data-tiering-enabled`.

Para Linux, macOS o Unix:

```
aws elasticache create-cache-cluster \
--cache-cluster-id my-cluster \
--cache-node-type cache.r4.large \
--engine redis \
--num-cache-nodes 1 \
--cache-parameter-group default.redis6.x \
--snapshot-arns arn:aws:s3:::amzn-s3-demo-bucket/snapshot.rdb
```

Para Windows:

```
aws elasticache create-cache-cluster ^
--cache-cluster-id my-cluster ^
--cache-node-type cache.r4.large ^
--engine redis ^
--num-cache-nodes 1 ^
```

```
--cache-parameter-group default.redis6.x ^
--snapshot-arns arn:aws:s3:::amzn-s3-demo-bucket/snapshot.rdb
```

Creación de un clúster de Valkey o Redis OSS (modo de clúster habilitado) (AWS CLI)

Los clústeres de Valkey o Redis OSS (modo de clúster habilitado) (API/CLI: grupos de replicación) no se pueden crear con la operación `create-cache-cluster`. Para crear un clúster de Valkey o Redis OSS (modo de clúster habilitado) (API/CLI: grupo de replicación), consulte Creación de un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado) desde cero (AWS CLI).

Para obtener más información, consulte la AWS CLI para el tema de referencia de ElastiCache `create-replication-group`.

Creación de un clúster para Valkey o Redis OSS (API de ElastiCache)

Para crear un clúster mediante la API de ElastiCache, utilice la acción `CreateCacheCluster`.

> **Important**
>
> Cuando su clúster esté disponible, se le cobrará por cada hora u hora parcial que el clúster esté activo, incluso si no lo está utilizando. Para dejar de incurrir en cargos por este clúster, debe eliminarlo. Consulte Eliminación de un clúster en ElastiCache.

Temas

- Creación de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) (API de ElastiCache)
- Creación de un clúster en Valkey o Redis OSS (modo de clúster habilitado) (API de ElastiCache)

Creación de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) (API de ElastiCache)

El siguiente código crea un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) (API de ElastiCache).

Se agregan saltos de línea para facilitar la lectura.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=CreateCacheCluster
    &CacheClusterId=my-cluster
```

```
    &CacheNodeType=cache.r4.large
    &CacheParameterGroup=default.redis3.2
    &Engine=redis
    &EngineVersion=3.2.4
    &NumCacheNodes=1
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &SnapshotArns.member.1=arn%3Aaws%3As3%3A%3A%3AmyS3Bucket%2Fdump.rdb
    &Timestamp=20150508T220302Z
    &Version=2015-02-02
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Credential=<credential>
    &X-Amz-Date=20150508T220302Z
    &X-Amz-Expires=20150508T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Signature=<signature>
```

Creación de un clúster en Valkey o Redis OSS (modo de clúster habilitado) (API de ElastiCache)

Los clústeres de Valkey o Redis OSS (modo de clúster habilitado) (API/CLI: grupos de replicación) no se pueden crear con la operación `CreateCacheCluster`. Para crear un clúster de Valkey o Redis OSS (modo de clúster habilitado) (API/CLI: grupo de replicación), consulte Creación de un grupo de replicación en Valkey o Redis OSS (modo de clúster habilitado) desde cero (API de ElastiCache).

Para obtener más información, consulte el tema de referencia de la API de ElastiCache `CreateReplicationGroup`.

Creación de un clúster para Memcached

En los siguientes ejemplos se muestra cómo crear un clúster con la Consola de administración de AWS, la AWS CLI y la API de ElastiCache.

Creación de un clúster de Memcached (consola)

Cuando se utiliza el motor de Memcached, Amazon ElastiCache admite la partición horizontal de los datos en varios nodos. Memcached permite la detección automática, por lo que no es necesario realizar un seguimiento de los puntos de enlace de cada nodo. Memcached realiza un seguimiento del punto de conexión de cada nodo, actualizando la lista de puntos de conexión a medida que se agregan y eliminan nodos. Todo lo que necesita su aplicación para interactuar con el clúster es el punto de conexión de configuración.

Para crear un clúster de Memcached mediante la consola, siga los pasos de Creación de un clúster de Valkey (modo de clúster deshabilitado) (consola). Cuando llegue al paso cinco, seleccione Crear caché de Memcached.

En cuanto el estado de su clúster sea available (disponible), podrá conceder a Amazon EC2 acceso a este, así como conectarse a él y comenzar a utilizarlo. Para obtener más información, consulte los pasos similares Paso 3. Autorización del acceso al clúster y Paso 4. Conexión al nodo de un clúster.

> **Important**
>
> Cuando su clúster esté disponible, se cobrará por cada hora u hora parcial que el clúster esté activo, incluso si no lo está utilizando de forma activa. Para dejar de incurrir en cargos por este clúster, debe eliminarlo. Consulte Eliminación de un clúster en ElastiCache.

Creación de un clúster (AWS CLI)

Para crear un clúster mediante la AWS CLI, utilice el comando `create-cache-cluster`.

> **Important**
>
> Cuando su clúster esté disponible, se cobrará por cada hora u hora parcial que el clúster esté activo, incluso si no lo está utilizando de forma activa. Para dejar de incurrir en cargos por este clúster, debe eliminarlo. Consulte Eliminación de un clúster en ElastiCache.

Creación de un clúster de caché de Memcached (AWS CLI)

El siguiente código de la CLI crea un clúster de Memcached con tres nodos.

Para Linux, macOS o Unix:

```
aws elasticache create-cache-cluster \
--cache-cluster-id my-cluster \
--cache-node-type cache.r4.large \
--engine memcached \
--engine-version 1.4.24 \
--cache-parameter-group default.memcached1.4 \
--num-cache-nodes 3
```

Para Windows:

```
aws elasticache create-cache-cluster ^
--cache-cluster-id my-cluster ^
--cache-node-type cache.r4.large ^
--engine memcached ^
--engine-version 1.4.24 ^
--cache-parameter-group default.memcached1.4 ^
--num-cache-nodes 3
```

Creación de un clúster para Memcached (API de ElastiCache)

Para crear un clúster mediante la API de ElastiCache, utilice la acción `CreateCacheCluster`.

> **Important**
>
> Cuando su clúster esté disponible, se le cobrará por cada hora u hora parcial que el clúster esté activo, incluso si no lo está utilizando. Para dejar de incurrir en cargos por este clúster, debe eliminarlo. Consulte Eliminación de un clúster en ElastiCache.

Temas

- Creación de un clúster de Memcached (API de ElastiCache)

Creación de un clúster de Memcached (API de ElastiCache)

El siguiente código crea un clúster de Memcached con tres nodos (API de ElastiCache).

Se agregan saltos de línea para facilitar la lectura.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=CreateCacheCluster
    &CacheClusterId=my-cluster
    &CacheNodeType=cache.r4.large
    &Engine=memcached
    &NumCacheNodes=3
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150508T220302Z
    &Version=2015-02-02
```

```
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Credential=<credential>
    &X-Amz-Date=20150508T220302Z
    &X-Amz-Expires=20150508T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Signature=<signature>
```

Visualización de los detalles de un clúster de ElastiCache

Puede consultar información detallada acerca de uno o varios clústeres mediante la consola de ElastiCache, la AWS CLI o la API de ElastiCache.

Visualización de los detalles de un clúster de Memcached (consola)

Puede ver los detalles de un clúster de Memcached con la consola de ElastiCache, la AWS CLI de ElastiCache o la API de ElastiCache.

El siguiente procedimiento detalla cómo consultar los detalles de un clúster de Memcached con la consola de ElastiCache.

Para ver los detalles de un clúster de Memcached

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En la lista de la esquina superior derecha, elija la región de AWS que le interese.
3. En el panel del motor de ElastiCache, elija Memcached. Al hacerlo, se muestra una lista de todos los clústeres que se ejecutan en el motor de Memcached.
4. Para ver los detalles de un clúster, elija el cuadro situado a la izquierda del nombre del clúster.
5. Para ver la información de los nodos, elija la pestaña Nodes (Nodos), que muestra información sobre el estado de los nodos y el punto de conexión.
6. Para ver las métricas, elija la pestaña Metrics (Métricas), que muestra las métricas pertinentes de todos los nodos del clúster. Para obtener más información, consulte Monitoreo del uso con métricas de CloudWatch
7. Elija la pestaña Network and security (Red y seguridad) para ver los detalles sobre la configuración de la conectividad de red y el grupo de subredes del clúster y el grupo de seguridad de VPC. Para obtener más información, consulte Subredes y grupos de subredes.
8. Elija la pestaña Maintenance (Mantenimiento) para ver los detalles en la configuración de mantenimiento del clúster. Para obtener más información, consulte Administración del mantenimiento de clústeres de ElastiCache.
9. Elija la pestaña Tags (Etiquetas) para ver los detalles de las etiquetas aplicadas a los recursos del clúster. Para obtener más información, consulte Etiquetado de los recursos de ElastiCache.

Visualización de los detalles de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) (consola)

Puede ver los detalles de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) con la consola de ElastiCache, la AWS CLI de ElastiCache o la API de ElastiCache.

El siguiente procedimiento detalla cómo consultar los detalles de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) con la consola de ElastiCache.

Visualización de los detalles de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel del motor de ElastiCache, elija Valkey o Redis OSS para ver una lista de todos los clústeres que se ejecutan en ese motor.
3. Para ver los detalles de un clúster, seleccione la casilla de verificación situada a la izquierda del nombre del clúster. Asegúrese de seleccionar un clúster que ejecute el motor de Valkey o Redis OSS, no Valkey en clúster ni Redis OSS en clúster. Al hacerlo se muestran los detalles sobre el clúster, incluido su punto de conexión principal.
4. Para ver la información de los nodos:
    a. Elija el nombre del clúster.
    b. Elija la pestaña Shards and nodes (Particiones y nodos). Se mostrarán los detalles de cada nodo, incluido el punto de conexión del nodo que tiene que usar para leer del clúster.
5. Para ver las métricas, elija la pestaña Metrics (Métricas), que muestra las métricas pertinentes de todos los nodos del clúster. Para obtener más información, consulte Monitoreo del uso con métricas de CloudWatch
6. Para ver los registros, elija la pestaña Logs (Registros), que indica si el clúster utiliza registros lentos o registros del motor y proporciona los detalles pertinentes. Para obtener más información, consulte Entrega de registro.
7. Elija la pestaña Network and security (Red y seguridad) para ver los detalles sobre la conectividad de red del clúster y la configuración del grupo de subredes. Para obtener más información, consulte Subredes y grupos de subredes.
8. Elija la pestaña Maintenance (Mantenimiento) para ver los detalles en la configuración de mantenimiento del clúster. Para obtener más información, consulte Administración del mantenimiento de clústeres de ElastiCache.

9. Elija la pestaña Service updates (Actualizaciones de servicio) para ver los detalles de las actualizaciones de servicio disponibles junto con la fecha de caducidad recomendada. Para obtener más información, consulte Actualizaciones de servicios en ElastiCache.
10. Elija la pestaña Tags (Etiquetas) para ver los detalles de las etiquetas aplicadas a los recursos del clúster. Para obtener más información, consulte Etiquetado de los recursos de ElastiCache.

Visualización de los detalles de un clúster de Valkey o Redis OSS (modo de clúster habilitado) (consola)

Puede ver los detalles de un clúster de Valkey o Redis OSS (modo de clúster habilitado) con la consola de ElastiCache, la AWS CLI de ElastiCache o la API de ElastiCache.

El siguiente procedimiento detalla cómo consultar los detalles de un clúster de Valkey o Redis OSS (modo de clúster habilitado) con la consola de ElastiCache.

Visualización de los detalles de un clúster de Valkey o Redis OSS (modo de clúster habilitado)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En la lista de la esquina superior derecha, elija la región de AWS que le interese.
3. En el panel del motor de ElastiCache, elija Valkey o Redis OSS para ver una lista de todos los clústeres que se ejecutan en ese motor.
4. Para ver los detalles de un clúster de Valkey o Redis OSS (modo de clúster habilitado), elija la casilla situada a la izquierda del nombre del clúster. Asegúrese de elegir un clúster que ejecute el motor de Valkey o Redis OSS en clúster.

   La pantalla se ampliará para mostrar el clúster y los detalles sobre el clúster, incluido el punto de conexión de configuración del clúster.
5. Para ver una lista de las particiones del clúster y el número de nodos en cada partición, elija la pestaña Shards and nodes (Particiones y nodos).
6. Para ver información específica sobre un nodo:

   - Elija el ID del fragmento.

     Al hacerlo, se muestra información sobre cada nodo, incluido el punto de conexión de cada nodo que debe utilizar para leer datos del clúster.

7. Para ver las métricas, elija la pestaña Metrics (Métricas), que muestra las métricas pertinentes de todos los nodos del clúster. Para obtener más información, consulte Monitoreo del uso con métricas de CloudWatch
8. Para ver los registros, elija la pestaña Logs (Registros), que indica si el clúster utiliza registros lentos o registros del motor y proporciona los detalles pertinentes. Para obtener más información, consulte Entrega de registro.
9. Elija la pestaña Network and security (Red y seguridad) para ver los detalles sobre la configuración de la conectividad de red y el grupo de subredes del clúster, el grupo de seguridad de la VPC y qué, si lo hay, método de cifrado está habilitado en el clúster. Para obtener más información, consulte Subredes y grupos de subredes y Seguridad de datos en Amazon ElastiCache.
10. Elija la pestaña Maintenance (Mantenimiento) para ver los detalles en la configuración de mantenimiento del clúster. Para obtener más información, consulte Administración del mantenimiento de clústeres de ElastiCache.
11. Elija la pestaña Service updates (Actualizaciones de servicio) para ver los detalles de las actualizaciones de servicio disponibles junto con la fecha de caducidad recomendada. Para obtener más información, consulte Actualizaciones de servicios en ElastiCache.
12. Elija la pestaña Tags (Etiquetas) para ver los detalles de las etiquetas aplicadas a los recursos del clúster. Para obtener más información, consulte Etiquetado de los recursos de ElastiCache.

Visualización de los detalles de un clúster de ElastiCache (AWS CLI)

El siguiente código muestra los detalles para *my-cluster*:

```
aws elasticache describe-cache-clusters --cache-cluster-id my-cluster
```

Sustituya *my-cluster* por el nombre del clúster en caso de que el clúster se cree con 1 nodo de caché y 0 particiones mediante el comando `create-cache-cluster`.

```
{
    "CacheClusters": [
        {
            "CacheClusterStatus": "available",
            "SecurityGroups": [
                {
                    "Status": "active",
                    "SecurityGroupId": "sg-dbe93fa2"
```

```
                }
            ],
            "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
            "Engine": "redis",
            "PreferredMaintenanceWindow": "wed:12:00-wed:13:00",
            "CacheSubnetGroupName": "default",
            "SnapshotWindow": "08:30-09:30",
            "TransitEncryptionEnabled": false,
            "AtRestEncryptionEnabled": false,
            "CacheClusterId": "my-cluster1",
            "CacheClusterCreateTime": "2018-02-26T21:06:43.420Z",
            "PreferredAvailabilityZone": "us-west-2c",
            "AuthTokenEnabled": false,
            "PendingModifiedValues": {},
            "CacheNodeType": "cache.r4.large",
           "DataTiering": "disabled",
            "CacheParameterGroup": {
                "CacheNodeIdsToReboot": [],
                "ParameterApplyStatus": "in-sync",
                "CacheParameterGroupName": "default.redis3.2"
            },
            "SnapshotRetentionLimit": 0,
            "AutoMinorVersionUpgrade": true,
            "EngineVersion": "3.2.10",
            "CacheSecurityGroups": [],
            "NumCacheNodes": 1
        }
```

```
{
    "CacheClusters": [
        {
            "SecurityGroups": [
                {
                    "Status": "active",
                    "SecurityGroupId": "sg-dbe93fa2"
                }
            ],
            "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
            "AuthTokenEnabled": false,
            "CacheSubnetGroupName": "default",
            "SnapshotWindow": "12:30-13:30",
```

```
            "AutoMinorVersionUpgrade": true,
            "CacheClusterCreateTime": "2018-02-26T21:13:24.250Z",
            "CacheClusterStatus": "available",
            "AtRestEncryptionEnabled": false,
            "PreferredAvailabilityZone": "us-west-2a",
            "TransitEncryptionEnabled": false,
            "ReplicationGroupId": "my-cluster2",
            "Engine": "redis",
            "PreferredMaintenanceWindow": "sun:08:30-sun:09:30",
            "CacheClusterId": "my-cluster2-001",
            "PendingModifiedValues": {},
            "CacheNodeType": "cache.r4.large",
            "DataTiering": "disabled",
            "CacheParameterGroup": {
                "CacheNodeIdsToReboot": [],
                "ParameterApplyStatus": "in-sync",
                "CacheParameterGroupName": "default.redis6.x"
            },
            "SnapshotRetentionLimit": 0,
            "EngineVersion": "6.0",
            "CacheSecurityGroups": [],
            "NumCacheNodes": 1
        },
        {
            "SecurityGroups": [
                {
                    "Status": "active",
                    "SecurityGroupId": "sg-dbe93fa2"
                }
            ],
            "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
            "AuthTokenEnabled": false,
            "CacheSubnetGroupName": "default",
            "SnapshotWindow": "12:30-13:30",
            "AutoMinorVersionUpgrade": true,
            "CacheClusterCreateTime": "2018-02-26T21:13:24.250Z",
            "CacheClusterStatus": "available",
            "AtRestEncryptionEnabled": false,
            "PreferredAvailabilityZone": "us-west-2b",
            "TransitEncryptionEnabled": false,
            "ReplicationGroupId": "my-cluster2",
            "Engine": "redis",
            "PreferredMaintenanceWindow": "sun:08:30-sun:09:30",
```

```
            "CacheClusterId": "my-cluster2-002",
            "PendingModifiedValues": {},
            "CacheNodeType": "cache.r4.large",
            "DataTiering": "disabled",
            "CacheParameterGroup": {
                "CacheNodeIdsToReboot": [],
                "ParameterApplyStatus": "in-sync",
                "CacheParameterGroupName": "default.redis6.x"
            },
            "SnapshotRetentionLimit": 0,
            "EngineVersion": "6.0",
            "CacheSecurityGroups": [],
            "NumCacheNodes": 1
        },
        {
            "SecurityGroups": [
                {
                    "Status": "active",
                    "SecurityGroupId": "sg-dbe93fa2"
                }
            ],
            "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
            "AuthTokenEnabled": false,
            "CacheSubnetGroupName": "default",
            "SnapshotWindow": "12:30-13:30",
            "AutoMinorVersionUpgrade": true,
            "CacheClusterCreateTime": "2018-02-26T21:13:24.250Z",
            "CacheClusterStatus": "available",
            "AtRestEncryptionEnabled": false,
            "PreferredAvailabilityZone": "us-west-2c",
            "TransitEncryptionEnabled": false,
            "ReplicationGroupId": "my-cluster2",
            "Engine": "redis",
            "PreferredMaintenanceWindow": "sun:08:30-sun:09:30",
            "CacheClusterId": "my-cluster2-003",
            "PendingModifiedValues": {},
            "CacheNodeType": "cache.r4.large",
            "DataTiering": "disabled",
            "CacheParameterGroup": {
                "CacheNodeIdsToReboot": [],
                "ParameterApplyStatus": "in-sync",
                "CacheParameterGroupName": "default.redis3.2"
            },
```

```
            "SnapshotRetentionLimit": 0,
            "EngineVersion": "3.2.10",
            "CacheSecurityGroups": [],
            "NumCacheNodes": 1
        }
```

```
{
    "CacheClusters": [
        {
            "SecurityGroups": [
                {
                    "Status": "active",
                    "SecurityGroupId": "sg-dbe93fa2"
                }
            ],
            "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
            "AuthTokenEnabled": true,
            "CacheSubnetGroupName": "default",
            "SnapshotWindow": "12:30-13:30",
            "AutoMinorVersionUpgrade": true,
            "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
            "CacheClusterStatus": "available",
            "AtRestEncryptionEnabled": true,
            "PreferredAvailabilityZone": "us-west-2a",
            "TransitEncryptionEnabled": true,
            "ReplicationGroupId": "my-cluster3",
            "Engine": "redis",
            "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
            "CacheClusterId": "my-cluster3-0001-001",
            "PendingModifiedValues": {},
            "CacheNodeType": "cache.r4.large",
            "DataTiering": "disabled",
            "CacheParameterGroup": {
                "CacheNodeIdsToReboot": [],
                "ParameterApplyStatus": "in-sync",
                "CacheParameterGroupName": "default.redis6.x.cluster.on"
            },
            "SnapshotRetentionLimit": 0,
            "EngineVersion": "6.0",
            "CacheSecurityGroups": [],
            "NumCacheNodes": 1
        },
```

```
        {
            "SecurityGroups": [
                {
                    "Status": "active",
                    "SecurityGroupId": "sg-dbe93fa2"
                }
            ],
            "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
            "AuthTokenEnabled": true,
            "CacheSubnetGroupName": "default",
            "SnapshotWindow": "12:30-13:30",
            "AutoMinorVersionUpgrade": true,
            "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
            "CacheClusterStatus": "available",
            "AtRestEncryptionEnabled": true,
            "PreferredAvailabilityZone": "us-west-2b",
            "TransitEncryptionEnabled": true,
            "ReplicationGroupId": "my-cluster3",
            "Engine": "redis",
            "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
            "CacheClusterId": "my-cluster3-0001-002",
            "PendingModifiedValues": {},
            "CacheNodeType": "cache.r4.large",
             "DataTiering": "disabled",
            "CacheParameterGroup": {
                "CacheNodeIdsToReboot": [],
                "ParameterApplyStatus": "in-sync",
                "CacheParameterGroupName": "default.redis3.2.cluster.on"
            },
            "SnapshotRetentionLimit": 0,
            "EngineVersion": "3.2.6",
            "CacheSecurityGroups": [],
            "NumCacheNodes": 1
        },
        {
            "SecurityGroups": [
                {
                    "Status": "active",
                    "SecurityGroupId": "sg-dbe93fa2"
                }
            ],
            "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
```

```
            "AuthTokenEnabled": true,
            "CacheSubnetGroupName": "default",
            "SnapshotWindow": "12:30-13:30",
            "AutoMinorVersionUpgrade": true,
            "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
            "CacheClusterStatus": "available",
            "AtRestEncryptionEnabled": true,
            "PreferredAvailabilityZone": "us-west-2c",
            "TransitEncryptionEnabled": true,
            "ReplicationGroupId": "my-cluster3",
            "Engine": "redis",
            "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
            "CacheClusterId": "my-cluster3-0001-003",
            "PendingModifiedValues": {},
            "CacheNodeType": "cache.r4.large",
             "DataTiering": "disabled",
            "CacheParameterGroup": {
                "CacheNodeIdsToReboot": [],
                "ParameterApplyStatus": "in-sync",
                "CacheParameterGroupName": "default.redis6.x.cluster.on"
            },
            "SnapshotRetentionLimit": 0,
            "EngineVersion": "6.0",
            "CacheSecurityGroups": [],
            "NumCacheNodes": 1
        },
        {
            "SecurityGroups": [
                {
                    "Status": "active",
                    "SecurityGroupId": "sg-dbe93fa2"
                }
            ],
            "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
            "AuthTokenEnabled": true,
            "CacheSubnetGroupName": "default",
            "SnapshotWindow": "12:30-13:30",
            "AutoMinorVersionUpgrade": true,
            "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
            "CacheClusterStatus": "available",
            "AtRestEncryptionEnabled": true,
            "PreferredAvailabilityZone": "us-west-2b",
            "TransitEncryptionEnabled": true,
```

```
            "ReplicationGroupId": "my-cluster3",
            "Engine": "redis",
            "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
            "CacheClusterId": "my-cluster3-0002-001",
            "PendingModifiedValues": {},
            "CacheNodeType": "cache.r4.large",
             "DataTiering": "disabled",
            "CacheParameterGroup": {
                "CacheNodeIdsToReboot": [],
                "ParameterApplyStatus": "in-sync",
                "CacheParameterGroupName": "default.redis6.x.cluster.on"
            },
            "SnapshotRetentionLimit": 0,
            "EngineVersion": "6.0",
            "CacheSecurityGroups": [],
            "NumCacheNodes": 1
        },
        {
            "SecurityGroups": [
                {
                    "Status": "active",
                    "SecurityGroupId": "sg-dbe93fa2"
                }
            ],
            "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
            "AuthTokenEnabled": true,
            "CacheSubnetGroupName": "default",
            "SnapshotWindow": "12:30-13:30",
            "AutoMinorVersionUpgrade": true,
            "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
            "CacheClusterStatus": "available",
            "AtRestEncryptionEnabled": true,
            "PreferredAvailabilityZone": "us-west-2c",
            "TransitEncryptionEnabled": true,
            "ReplicationGroupId": "my-cluster3",
            "Engine": "redis",
            "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
            "CacheClusterId": "my-cluster3-0002-002",
            "PendingModifiedValues": {},
            "CacheNodeType": "cache.r4.large",
             "DataTiering": "disabled",
            "CacheParameterGroup": {
                "CacheNodeIdsToReboot": [],
```

```
                "ParameterApplyStatus": "in-sync",
                "CacheParameterGroupName": "default.redis3.2.cluster.on"
            },
            "SnapshotRetentionLimit": 0,
            "EngineVersion": "3.2.6",
            "CacheSecurityGroups": [],
            "NumCacheNodes": 1
        },
        {
            "SecurityGroups": [
                {
                    "Status": "active",
                    "SecurityGroupId": "sg-dbe93fa2"
                }
            ],
            "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
            "AuthTokenEnabled": true,
            "CacheSubnetGroupName": "default",
            "SnapshotWindow": "12:30-13:30",
            "AutoMinorVersionUpgrade": true,
            "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
            "CacheClusterStatus": "available",
            "AtRestEncryptionEnabled": true,
            "PreferredAvailabilityZone": "us-west-2a",
            "TransitEncryptionEnabled": true,
            "ReplicationGroupId": "my-cluster3",
            "Engine": "redis",
            "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
            "CacheClusterId": "my-cluster3-0002-003",
            "PendingModifiedValues": {},
            "CacheNodeType": "cache.r4.large",
             "DataTiering": "disabled",
            "CacheParameterGroup": {
                "CacheNodeIdsToReboot": [],
                "ParameterApplyStatus": "in-sync",
                "CacheParameterGroupName": "default.redis6.x.cluster.on"
            },
            "SnapshotRetentionLimit": 0,
            "EngineVersion": "6.0",
            "CacheSecurityGroups": [],
            "NumCacheNodes": 1
        }
    ]
```

```
}
```

En un caso en el que el clúster se cree mediante la Consola de administración de AWS (nodo de clúster habilitado o desactivado con 1 o más particiones), utilice el siguiente comando para describir los detalles del clúster (sustituya *my-cluster* por el nombre del grupo de replicación (nombre del clúster):

```
aws elasticache describe-replication-groups --replication-group-id my-cluster
```

Para obtener más información, consulte la AWS CLI para el tema de ElastiCache describe-cache-clusters.

Visualización de los detalles de un clúster de ElastiCache (API de ElastiCache)

Puede ver los detalles de un clúster con la acción `DescribeCacheClusters` de la API de ElastiCache. Si se incluye el parámetro `CacheClusterId`, se devolverán detalles del clúster especificado. Si el parámetro `CacheClusterId` se omite, se devolverán los detalles de hasta `MaxRecords` clústeres (el valor predeterminado es 100). El valor de `MaxRecords` no puede ser inferior a 20 ni superior a 100.

El siguiente código enumera los detalles de `my-cluster`.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DescribeCacheClusters
   &CacheClusterId=my-cluster
   &Version=2015-02-02
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &X-Amz-Credential=<credential>
```

El siguiente código enumera los detalles de hasta 25 clústeres.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DescribeCacheClusters
   &MaxRecords=25
   &Version=2015-02-02
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
```

```
    &X-Amz-Credential=<credential>
```

Para obtener más información, consulte el tema de referencia de la API de ElastiCache `DescribeCacheClusters`.

Modificación de un clúster de ElastiCache

Además de añadir o quitar nodos de un clúster de ElastiCache, puede que haya veces en las que necesite realizar otros cambios, como, por ejemplo, añadir un grupo de seguridad o cambiar el periodo de mantenimiento o un grupo de parámetros.

Recomendamos que el periodo de mantenimiento corresponda al momento de mínimo uso. Esto puede requerir alguna modificación de vez en cuando.

Cuando se modifican los parámetros de un clúster, el cambio se aplica al clúster inmediatamente o después de reiniciarlo. Esto es cierto tanto si se modifica el propio grupo de parámetros del clúster como si se modifica el valor de un parámetro del grupo. Para determinar cuándo se aplica un cambio de parámetros determinado, consulte la sección Aplicación de los cambios de la columna Detalles en las tablas para Parámetros específicos de Memcached y Parámetros de Valkey y Redis OSS. Para obtener información acerca de cómo reiniciar los nodos de un clúster, consulte Reinicio de nodos.

Uso de la Consola de administración de AWS de ElastiCache

Pasos para modificar un clúster

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En la lista de la esquina superior derecha, elija la región de AWS en la que se encuentra el clúster que desea modificar.
3. En el panel de navegación, elija el motor que se ejecuta en el clúster que desea modificar.

   Se mostrará una lista de los clústeres del motor elegido.
4. En la lista de clústeres, para el clúster que desea modificar, elija su nombre.
5. Elija Actions (Acciones) y después Modify (Modificar).

   Aparecerá la ventana Modify Cluster.
6. En la ventana Modificar clúster, haga las modificaciones que desee. Las opciones son:

   - Descripción
   - Modo de clúster: Para modificar el modo de clúster de desactivado a habilitado, primero debe establecer el modo de clúster en compatible.

     El modo compatible permite que los clientes de Valkey o Redis OSS se conecten mediante el modo clúster habilitado y el modo clúster desactivado. Después de migrar todos los

clientes de Valkey o Redis OSS para usar el modo de clúster habilitado, puede completar la configuración del modo de clúster y establecer el modo de clúster en Habilitado.

- Compatibilidad de la versión del motor

> **Important**
>
> Puede actualizar a versiones de motor más recientes. Si actualiza las versiones principales del motor, por ejemplo de 5.0.6 a 6.0, debe seleccionar una familia de grupos de parámetros que sea compatible con la versión del motor nueva. Para obtener más información sobre cómo hacerlo, consulte Administración de versiones para ElastiCache. Sin embargo, no se puede volver a versiones del motor anteriores, salvo si se elimina el clúster existente y se vuelve a crear.

- Grupos de seguridad de VPC.
- Grupo de parámetros
- Tipo de nodo

> **Note**
>
> Si el clúster utiliza un tipo de nodo de la familia r6gd, solo puede elegir un tamaño de nodo diferente dentro de esa familia. Si elige un tipo de nodo de la familia r6gd, la organización de datos en niveles se activará automáticamente. Para obtener más información, consulte Organización de datos en niveles.

- Multi-AZ
- Conmutación por error automática (solo modo de clúster deshabilitado)
- Habilitar copias de seguridad automáticas
- ID del nodo de copia de seguridad
- Backup Retention Period
- Periodo de copia de seguridad
- Tema para la notificación de SNS

- Compatibilidad de la versión del motor de Memcached
- Tipo de red

> **Note**
>
> Si va a cambiar de IPv4 a IPv6, debe seleccionar o crear grupos de subredes compatibles con IPv6. Para obtener más información, consulte Elección de un tipo de red en ElastiCache.

- Grupos de seguridad de VPC.
- Grupo de parámetros
- Periodo de mantenimiento
- Tema para la notificación de SNS

El cuadro Aplicar inmediatamente solo se aplica a las modificaciones del tipo de nodo y de versión del motor. Para aplicar los cambios inmediatamente, active la casilla Apply Immediately (Aplicar inmediatamente). Si no se elige esta casilla, las modificaciones en la versión del motor se aplicarán durante el próximo periodo de mantenimiento. Otras modificaciones, como el cambio del periodo de mantenimiento, se aplican inmediatamente.

Habilitación/deshabilitación de la entrega de registros para Redis

1. En la lista de clústeres, elija el clúster que desea modificar. Elija el Cluster name (Nombre del clúster) y no la casilla de verificación al lado.
2. En la página Detalles del clúster, seleccione la pestaña Registros.
3. Para habilitar/deshabilitar los registros lentos, elija Habilitar o Deshabilitar.

   Si elige habilitar:

   a. En Log format (Formato de registro), elija Text (Texto) o JSON.

   b. En Log destination Type (Tipo de destino de registro), elija CloudWatch Logs o Kinesis Firehose.

   c. En Destino del registro, elija Crear nuevo e introduzca el nombre del grupo de registro de Registros de CloudWatch o el nombre del flujo de Kinesis Data Firehose. O elija Seleccionar existente y, a continuación, elija el nombre del grupo de registros de Registros de CloudWatch o el nombre del flujo de Kinesis Data Firehose.

   d. Seleccione Habilitar.

Cambio de su configuración para Redis

1. Elija Modificar.
2. En Log format (Formato de registro), elija Text (Texto) o JSON.
3. En Destination Type (Tipo de destino), elija CloudWatch Logs o Kinesis Firehose.
4. En Log destination (Destino del registro), elija Create new (Crear nuevo) e ingrese el nombre del grupo de registro de CloudWatchLogs o el nombre de la secuencia de Kinesis Data Firehose. O elija Select existing (Seleccionar existente) y, a continuación, elija el nombre del grupo de registro de CloudWatchLogs o el nombre de la secuencia de Kinesis Data Firehose.

Uso de la AWS CLI con ElastiCache

Puede modificar un clúster existente con la operación AWS CLI de `modify-cache-cluster`. Para modificar un valor de configuración de un clúster, especifique el ID del clúster, el parámetro que desea cambiar y el nuevo valor del parámetro. El siguiente ejemplo cambia el periodo de mantenimiento de un clúster denominado `my-cluster` y aplica el cambio inmediatamente.

> **Important**
>
> Puede actualizar a versiones del motor más recientes de Memcached. Para obtener más información sobre cómo hacerlo, consulte Administración de versiones para ElastiCache. Sin embargo, no se puede volver a versiones del motor anteriores, salvo si se elimina el clúster existente y se vuelve a crear.

> **Important**
>
> Puede actualizar a versiones del motor más recientes de Valkey o Redis OSS. Si actualiza las versiones principales del motor, por ejemplo de Redis OSS 5.0.6 a Redis OSS 6.0, debe seleccionar una familia de grupos de parámetros que sea compatible con la versión del motor nueva. Para obtener más información sobre cómo hacerlo, consulte Administración de versiones para ElastiCache. Sin embargo, no se puede volver a versiones del motor anteriores, salvo si se elimina el clúster existente y se vuelve a crear.

Para Linux, macOS o Unix:

```
aws elasticache modify-cache-cluster \
    --cache-cluster-id my-cluster \
    --preferred-maintenance-window sun:23:00-mon:02:00
```

Para Windows:

```
aws elasticache modify-cache-cluster ^
    --cache-cluster-id my-cluster ^
    --preferred-maintenance-window sun:23:00-mon:02:00
```

El parámetro `--apply-immediately` solo se aplica a modificaciones en el tipo de nodo, la versión del motor y el número de nodos de un clúster. Si desea aplicar cualquiera de estos cambios de forma inmediata, utilice el parámetro `--apply-immediately`. Si prefiere aplazar estos cambios al siguiente periodo de mantenimiento, utilice el parámetro `--no-apply-immediately`. Otras modificaciones, como el cambio del periodo de mantenimiento, se aplican inmediatamente.

Para obtener más información, consulte la AWS CLI para el tema de ElastiCache `modify-cache-cluster`.

Uso de la API de ElastiCache

Puede modificar un clúster existente con la operación `ModifyCacheCluster` de la API de ElastiCache. Para modificar un valor de configuración de un clúster, especifique el ID del clúster, el parámetro que desea cambiar y el nuevo valor del parámetro. El siguiente ejemplo cambia el periodo de mantenimiento de un clúster denominado `my-cluster` y aplica el cambio inmediatamente.

Important

Puede actualizar a versiones del motor más recientes de Memcached. Para obtener más información sobre cómo hacerlo, consulte Administración de versiones para ElastiCache. Sin embargo, no se puede volver a versiones del motor anteriores, salvo si se elimina el clúster existente y se vuelve a crear.

Important

Puede actualizar a versiones del motor más recientes de Valkey o Redis OSS. Si actualiza las versiones principales del motor, por ejemplo de Redis OSS 5.0.6 a Redis OSS 6.0, debe seleccionar una familia de grupos de parámetros que sea compatible con la versión del

motor nueva. Para obtener más información sobre cómo hacerlo, consulte Administración de versiones para ElastiCache. Sin embargo, no se puede volver a versiones del motor anteriores, salvo si se elimina el clúster existente y se vuelve a crear.

Se agregan saltos de línea para facilitar la lectura.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=ModifyCacheCluster
    &CacheClusterId=my-cluster
    &PreferredMaintenanceWindow=sun:23:00-mon:02:00
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150901T220302Z
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20150202T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20150901T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

El parámetro `ApplyImmediately` solo se aplica a modificaciones en el tipo de nodo, la versión del motor y el número de nodos de un clúster. Si desea aplicar cualquiera de estos cambios de forma inmediata, establezca el parámetro `ApplyImmediately` en `true`. Si prefiere aplazar estos cambios al siguiente periodo de mantenimiento, establezca el parámetro `ApplyImmediately` en `false`. Otras modificaciones, como el cambio del periodo de mantenimiento, se aplican inmediatamente.

Para obtener más información, consulte el tema de referencia de la API de ElastiCache `ModifyCacheCluster`.

Adición de nodos a un clúster de ElastiCache

Si se agregan nodos a un clúster de Memcached, se incrementa el número de particiones del clúster. Cuando se modifica el número de particiones de un clúster, algunos de los espacios de claves deben reasignarse para que se correspondan con el nodo adecuado. Si los espacios de clave se reasignan temporalmente, aumentará el número de errores de caché en el clúster. Para obtener más información, consulte Configuración de su cliente de ElastiCache para un equilibrio de carga eficaz (Memcached).

Para volver a configurar el clúster de Valkey o Redis OSS (modo de clúster habilitado), consulte Escalado de clústeres de Valkey o Redis OSS (modo de clúster habilitado).

Puede utilizar la consola de administración de ElastiCache, la AWS CLI o la API de ElastiCache para agregar nodos al clúster.

Uso de la Consola de administración de AWS de ElastiCache

Si desea añadir un nodo a un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) de un solo nodo (uno sin la replicación habilitada), el proceso conlleva dos pasos: en primer lugar, añadir la replicación y, a continuación, añadir un nodo de réplica.

Temas

- Adición de replicación a un clúster de Valkey o Redis OSS sin particiones
- Adición de nodos a un clúster de ElastiCache (consola)

El siguiente procedimiento añade la replicación a un clúster de Valkey o Redis OSS de un solo nodo sin la replicación habilitada. Al agregar la reproducción, el nodo existente se convierte en el nodo principal en el clúster con la reproducción habilitada. Tras agregar la reproducción, puede agregar hasta 5 nodos de réplica al clúster.

Adición de replicación a un clúster de Valkey o Redis OSS sin particiones

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Clústeres de Valkey o Clústeres de Redis OSS.

   Se mostrará una lista de clústeres que ejecutan el motor.

3. Elija el nombre de un clúster al que desea agregar nodos, pero no la casilla situada a la izquierda del nombre del clúster.

   En un clúster de Redis OSS que no tiene la replicación habilitada, se dan las siguientes condiciones:

   - Ejecuta Redis OSS, no Redis OSS en clúster.
   - No tiene ningún fragmento.

     Si el clúster tiene algún fragmento, la reproducción ya está habilitada y podrá continuar con Adición de nodos a un clúster de ElastiCache (consola).
4. Seleccione Add replication (Agregar reproducción).
5. En Agregar reproducción, escriba una descripción para este clúster con la reproducción habilitada.
6. Elija Agregar.

   Cuando el estado del clúster vuelva a ser available, podrá continuar con el siguiente procedimiento y agregar réplicas al clúster.

Adición de nodos a un clúster de ElastiCache (consola)

El siguiente procedimiento se puede utilizar para agregar nodos a un clúster.

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija el motor que se ejecuta en el clúster en el que desea agregar nodos.

   Se mostrará una lista de los clústeres que ejecutan el motor elegido.
3. En la lista de clústeres, para el clúster al que desea agregar un nodo, elija su nombre.

   Si el clúster es un clúster de Valkey o Redis OSS (modo de clúster habilitado), consulte Escalado de clústeres de Valkey o Redis OSS (modo de clúster habilitado).

   Si el clúster es un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) sin particiones, en primer lugar, complete los pasos de Adición de replicación a un clúster de Valkey o Redis OSS sin particiones.
4. Elija Add node (Agregar nodo).

5. Rellene la información que se solicita en el cuadro de diálogo Add Node (Agregar nodo).
6. Seleccione el botón Apply Immediately - Yes (Aplicar inmediatamente: sí) para aplicar este cambio de inmediato o seleccione No para aplazar el cambio hasta el próximo periodo de mantenimiento del clúster.

   Impacto de las nuevas solicitudes de adición y eliminación en las solicitudes pendientes

Escenarios	Operación pendiente	Solicitud nueva	Resultados
Escenario 1	Eliminar	Eliminar	La nueva solicitud de eliminación, pendiente o inmediata, reemplaza a la solicitud de eliminación pendiente. Por ejemplo, si los nodos 0001, 0003 y 0007 se encuentran pendientes de eliminación y se emite una solicitud nueva para eliminar los nodos 0002 y 0004, solo se eliminarán los nodos 0002 y 0004. Los nodos 0001, 0003 y 0007 no se eliminarán.
Escenario 2	Eliminar	Creación	La nueva solicitud de creación, pendiente o inmediata, reemplaza a la solicitud de eliminación pendiente. Por ejemplo, si los nodos 0001, 0003 y 0007 se encuentran pendientes de eliminación y se emite una solicitud nueva para crear un nodo, se creará un nodo nuevo y no se eliminarán los nodos 0001, 0003 y 0007.
Escenario 3	Creación	Eliminar	La nueva solicitud de eliminación, pendiente o inmediata, reemplaza a la solicitud de creación pendiente. Por ejemplo, si hay una solicitud pendiente para crear dos nodos y se emite una solicitud nueva para eliminar el nodo 0003, no se crearán nodos nuevos y se eliminará el nodo 0003.

Escenarios	Operación pendiente	Solicitud nueva	Resultados
Escenario 4	Creación	Creación	La nueva solicitud de creación se agrega a la solicitud de creación pendiente. Por ejemplo, si hay una solicitud pendiente para crear dos nodos y se emite una solicitud nueva para crear tres nodos, las solicitudes nuevas se agregarán a la solicitud pendiente y se crearán cinco nodos. **Important** Si la nueva solicitud de creación se establece en Apply Immediately - Yes (Aplicar de inmediato: sí), todas las solicitudes de creación se ejecutan de inmediato. Si la nueva solicitud de creación se establece en Apply Immediately - No (Aplicar de inmediato: no), todas las solicitudes de creación quedarán pendientes.

Para determinar qué operaciones se encuentran pendientes, elija la pestaña de Description (Descripción) y verifique cuántas creaciones o eliminaciones pendientes se muestran. No puede tener creaciones y eliminaciones pendientes al mismo tiempo.

7. Seleccione el botón Add (Agregar).

   Después de unos minutos, los nuevos nodos deberían aparecer en la lista de nodos con el estado creating. Si no aparecen, actualice página del navegador. Cuando el estado cambie a nodo disponible el nuevo nodo puede usarse.

Uso de la AWS CLI con ElastiCache

Para agregar nodos a un clúster utilizando la AWS CLI, use la operación AWS CLI de la `modify-cache-cluster` con los parámetros siguientes:

- `--cache-cluster-id` El ID del clúster al que desea agregar nodos.
- `--num-cache-nodes` El parámetro `--num-cache-nodes` especifica el número de nodos que desea en este clúster una vez aplicada la modificación. Para añadir nodos a este clúster, el valor de `--num-cache-nodes` debe ser mayor que el número actual de nodos en este clúster. Si este valor es menor que el número actual de nodos, ElastiCache espera el parámetro `cache-node-ids-to-remove` y una lista de nodos que quitar del clúster. Para obtener más información, consulte Uso de la AWS CLI con ElastiCache.
- `--apply-immediately` o `--no-apply-immediately`, que especifica si desea añadir estos nodos de forma inmediata o durante el próximo periodo de mantenimiento.

Para Linux, macOS o Unix:

```
aws elasticache modify-cache-cluster \
    --cache-cluster-id my-cluster \
    --num-cache-nodes 5 \
    --apply-immediately
```

Para Windows:

```
aws elasticache modify-cache-cluster ^
    --cache-cluster-id my-cluster ^
    --num-cache-nodes 5 ^
    --apply-immediately
```

Esta operación produce un resultado similar al siguiente (formato JSON):

```
{
    "CacheCluster": {
        "Engine": "memcached",
        "CacheParameterGroup": {
            "CacheNodeIdsToReboot": [],
            "CacheParameterGroupName": "default.memcached1.4",
            "ParameterApplyStatus": "in-sync"
        },
        "CacheClusterId": "my-cluster",
        "PreferredAvailabilityZone": "us-west-2b",
        "ConfigurationEndpoint": {
            "Port": 11211,
            "Address": "rlh-mem000.7alc7bf-example.cfg.usw2.cache.amazonaws.com"
```

```
        },
        "CacheSecurityGroups": [],
        "CacheClusterCreateTime": "2016-09-21T16:28:28.973Z",
        "AutoMinorVersionUpgrade": true,
        "CacheClusterStatus": "modifying",
        "NumCacheNodes": 2,
        "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
        "SecurityGroups": [
            {
                "Status": "active",
                "SecurityGroupId": "sg-dbe93fa2"
            }
        ],
        "CacheSubnetGroupName": "default",
        "EngineVersion": "1.4.24",
        "PendingModifiedValues": {
            "NumCacheNodes": 5
        },
        "PreferredMaintenanceWindow": "sat:09:00-sat:10:00",
        "CacheNodeType": "cache.m3.medium",
         "DataTiering": "disabled",
    }
}
```

Para obtener más información, consulte el tema AWS CLI de la `modify-cache-cluster`.

Uso de la AWS CLI con ElastiCache

Si desea añadir nodos a un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) existente que no tiene la replicación habilitada, primero debe crear el grupo de replicación y establecer el clúster existente como principal. Para obtener más información, consulte Creación de un grupo de replicación mediante un clúster de Valkey o Redis OSS disponible (AWS CLI). Una vez que el estado del grupo de replicación sea available, puede continuar con el siguiente proceso.

Para agregar nodos a un clúster utilizando la AWS CLI, use la operación AWS CLI de la `increase-replica-count` con los parámetros siguientes:

- `--replication-group-id`: el ID del grupo de reproducción al que desea agregar nodos.
- `--new-replica-count` especifica el número de nodos que desea en este grupo de reproducción una vez que se aplica la modificación. Para añadir nodos a este clúster, el valor de `--new-replica-count` debe ser mayor que el número actual de nodos en este clúster.

- `--apply-immediately` o `--no-apply-immediately`, que especifica si desea añadir estos nodos de forma inmediata o durante el próximo periodo de mantenimiento.

Para Linux, macOS o Unix:

```
aws elasticache increase-replica-count \
    --replication-group-id my-replication-group \
    --new-replica-count 4 \
    --apply-immediately
```

Para Windows:

```
aws elasticache increase-replica-count ^
    --replication-group-id my-replication-group ^
    --new-replica-count 4 ^
    --apply-immediately
```

Esta operación produce un resultado similar al siguiente (formato JSON):

```
{
    "ReplicationGroup": {
        "ReplicationGroupId": "node-test",
        "Description": "node-test",
        "Status": "modifying",
        "PendingModifiedValues": {},
        "MemberClusters": [
            "node-test-001",
            "node-test-002",
            "node-test-003",
            "node-test-004",
            "node-test-005"
        ],
        "NodeGroups": [
            {
                "NodeGroupId": "0001",
                "Status": "modifying",
                "PrimaryEndpoint": {
                    "Address": "node-test.zzzzzz.ng.0001.usw2.cache.amazonaws.com",
                    "Port": 6379
                },
                "ReaderEndpoint": {
                    "Address": "node-test.zzzzzz.ng.0001.usw2.cache.amazonaws.com",
```

```
                    "Port": 6379
                },
                "NodeGroupMembers": [
                    {
                        "CacheClusterId": "node-test-001",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address": "node-
test-001.zzzzzz.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2a",
                        "CurrentRole": "primary"
                    },
                    {
                        "CacheClusterId": "node-test-002",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address": "node-
test-002.zzzzzz.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2c",
                        "CurrentRole": "replica"
                    },
                    {
                        "CacheClusterId": "node-test-003",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address": "node-
test-003.zzzzzz.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2b",
                        "CurrentRole": "replica"
                    }
                ]
            }
        ],
        "SnapshottingClusterId": "node-test-002",
        "AutomaticFailover": "enabled",
        "MultiAZ": "enabled",
        "SnapshotRetentionLimit": 1,
        "SnapshotWindow": "07:30-08:30",
```

```
        "ClusterEnabled": false,
        "CacheNodeType": "cache.r5.large",
         "DataTiering": "disabled",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false,
        "ARN": "arn:aws:elasticache:us-west-2:123456789012:replicationgroup:node-test"
    }
}
```

Para obtener más información, consulte el tema AWS CLI de la `increase-replica-count`.

Uso de la API de ElastiCache

Si desea añadir nodos a un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) existente que no tiene la replicación habilitada, primero debe crear el grupo de replicación y establecer el clúster existente como principal. Para obtener más información, consulte Adición de réplicas a un clúster independiente de Valkey o Redis OSS (modo de clúster deshabilitado) (API de ElastiCache). Una vez que el estado del grupo de replicación sea available, puede continuar con el siguiente proceso.

Para agregar nodos a un clúster (API de ElastiCache)

- Realice una llamada a la operación API `IncreaseReplicaCount` con los parámetros siguientes:
  - `ReplicationGroupId` El ID del clúster al que desea agregar nodos.
  - `NewReplicaCount` El parámetro `NewReplicaCount` especifica el número de nodos que desea en este clúster una vez aplicada la modificación. Para añadir nodos a este clúster, el valor de `NewReplicaCount` debe ser mayor que el número actual de nodos en este clúster. Si este valor es menor que el número actual de nodos, utilice la API `DecreaseReplicaCount` con el número de nodos que se debe quitar del clúster.
  - `ApplyImmediately` especifica si desea añadir estos nodos de forma inmediata o durante el próximo periodo de mantenimiento.
  - `Region` especifica la región de AWS del clúster al que desea agregar nodos.

  En el siguiente ejemplo se muestra una llamada para añadir nodos a un clúster.

Example

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=IncreaseReplicaCount
    &ApplyImmediately=true
    &NumCacheNodes=4
    &ReplicationGroupId=my-replication-group
    &Region=us-east-2
    &Version=2014-12-01
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20141201T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Para obtener más información, consulte el tema de la API de ElastiCache `IncreaseReplicaCount`.

Uso de la API de ElastiCache

Para agregar nodos a un clúster (API de ElastiCache)

- Realice una llamada a la operación API `ModifyCacheCluster` con los parámetros siguientes:
  - `CacheClusterId` El ID del clúster al que desea agregar nodos.
  - `NumCacheNodes` El parámetro `NumCachNodes` especifica el número de nodos que desea en este clúster una vez aplicada la modificación. Para añadir nodos a este clúster, el valor de `NumCacheNodes` debe ser mayor que el número actual de nodos en este clúster. Si este valor es menor que el número actual de nodos, ElastiCache espera el parámetro `CacheNodeIdsToRemove` con una lista de nodos que quitar del clúster (consulte Uso de la API de ElastiCache con Memcached).
  - `ApplyImmediately` especifica si desea añadir estos nodos de forma inmediata o durante el próximo periodo de mantenimiento.
  - `Region` especifica la región de AWS del clúster al que desea agregar nodos.

En el siguiente ejemplo se muestra una llamada para añadir nodos a un clúster.

Example

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=ModifyCacheCluster
    &ApplyImmediately=true
    &NumCacheNodes=5
 &CacheClusterId=my-cluster
 &Region=us-east-2
    &Version=2014-12-01
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20141201T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Para obtener más información, consulte el tema de la API de ElastiCache `ModifyCacheCluster`.

Eliminación de nodos de un clúster de ElastiCache

Para eliminar un nodo de un clúster de Valkey, Memcached o Redis OSS, puede utilizar la Consola de administración de AWS, la AWS CLI o la API de ElastiCache.

> **Note**
>
> Cada vez que cambie el número de nodos de un clúster de Memcached, debe volver a asignar al menos algunos de su espacio de claves para que se asocien al nodo correcto. Para obtener información más detallada sobre cómo equilibrar la carga de un clúster de Memcached, consulte Configuración de su cliente de ElastiCache para un equilibrio de carga eficaz (Memcached).

Uso de la Consola de administración de AWS de ElastiCache

Para eliminar nodos de un clúster (consola)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En la lista de la esquina superior derecha, elija la región de AWS del clúster del que desea quitar nodos.
3. En el panel de navegación, elija el motor que se ejecuta en el clúster del que desea quitar un nodo.

   Se mostrará una lista de los clústeres que ejecutan el motor elegido.
4. En la lista de clústeres, elija el nombre del clúster del que desea quitar un nodo.

   Aparecerá una lista de nodos del clúster.
5. Elija la casilla situada a la izquierda del ID del nodo que desee quitar. Con la consola de ElastiCache, solo puede eliminar un nodo a la vez, por lo que elegir varios nodos significa que no puede utilizar el botón Delete node (Eliminar nodo).

   Aparecerá la página Eliminar nodo.
6. Para eliminar el nodo, complete la página Eliminar nodo y elija Eliminar nodo. Para mantener el nodo, elija Cancelar.

> **Important**
>
> Con Valkey o Redis OSS, si va a eliminar los resultados de los nodos del clúster que ya no son compatibles con multi-AZ, asegúrese de desactivar primero la casilla de verificación Multi-AZ y, a continuación, elimine el nodo. Si desactiva la casilla de verificación Multi-AZ puede elegir habilitar la Conmutación por error automática.

Impacto de las nuevas solicitudes de adición y eliminación en las solicitudes pendientes

Escenarios	Operación pendiente	Solicitud nueva	Resultados
Escenario 1	Eliminar	Eliminar	La nueva solicitud de eliminación, pendiente o inmediata, reemplaza a la solicitud de eliminación pendiente. Por ejemplo, si los nodos 0001, 0003 y 0007 se encuentran pendientes de eliminación y se emite una solicitud nueva para eliminar los nodos 0002 y 0004, solo se eliminarán los nodos 0002 y 0004. Los nodos 0001, 0003 y 0007 no se eliminarán.
Escenario 2	Eliminar	Creación	La nueva solicitud de creación, pendiente o inmediata, reemplaza a la solicitud de eliminación pendiente. Por ejemplo, si los nodos 0001, 0003 y 0007 se encuentran pendientes de eliminación y se emite una solicitud nueva para crear un nodo, se creará un nodo nuevo y no se eliminarán los nodos 0001, 0003 y 0007.
Escenario 3	Creación	Eliminar	La nueva solicitud de eliminación, pendiente o inmediata, reemplaza a la solicitud de creación pendiente. Por ejemplo, si hay una solicitud pendiente para crear dos nodos y se emite una solicitud nueva para

Escenarios	Operación pendiente	Solicitud nueva	Resultados
			eliminar el nodo 0003, no se crearán nodos nuevos y se eliminará el nodo 0003.
Escenario 4	Creación	Creación	La nueva solicitud de creación se agrega a la solicitud de creación pendiente. Por ejemplo, si hay una solicitud pendiente para crear dos nodos y se emite una solicitud nueva para crear tres nodos, las solicitudes nuevas se agregarán a la solicitud pendiente y se crearán cinco nodos. **Important** Si la nueva solicitud de creación se establece en Apply Immediately - Yes (Aplicar de inmediato: sí), todas las solicitudes de creación se ejecutan de inmediato. Si la nueva solicitud de creación se establece en Apply Immediately - No (Aplicar de inmediato: no), todas las solicitudes de creación quedarán pendientes.

Para determinar qué operaciones se encuentran pendientes, elija la pestaña de Description (Descripción) y verifique cuántas creaciones o eliminaciones pendientes se muestran. No puede tener creaciones y eliminaciones pendientes al mismo tiempo.

Uso de la AWS CLI con ElastiCache

1. Especifique los ID de los nodos que desee quitar. Para obtener más información, consulte Visualización de los detalles de un clúster de ElastiCache.
2. Utilice la operación `decrease-replica-count` de la CLI con una lista de los nodos que desea quitar, como en el siguiente ejemplo.

Para quitar nodos de un clúster a través de la interfaz de línea de comandos, utilice el comando `decrease-replica-count` con los siguientes parámetros:

- `--replication-group-id`: el ID del grupo de reproducción del que desea eliminar nodos.
- `--new-replica-count` El parámetro `--new-replica-count` especifica el número de nodos que desea en este clúster una vez aplicada la modificación.
- `--replicas-to-remove` Una lista de los ID de nodos que desea quitar de este clúster.
- `--apply-immediately` o `--no-apply-immediately` especifican si desea quitar estos nodos de forma inmediata o durante el próximo periodo de mantenimiento.
- `--region` especifica la región de AWS del clúster del que desea quitar nodos.

> **Note**
>
> Solo puede pasar uno de los parámetros `--replicas-to-remove` o `--new-replica-count` al llamar a esta operación.

Para Linux, macOS o Unix:

```
aws elasticache decrease-replica-count \
    --replication-group-id my-replication-group \
    --new-replica-count 2 \
    --region us-east-2 \
    --apply-immediately
```

Para Windows:

```
aws elasticache decrease-replica-count ^
    --replication-group-id my-replication-group ^
    --new-replica-count 3 ^
    --region us-east-2 ^
    --apply-immediately
```

Esta operación produce un resultado similar al siguiente (formato JSON):

```
{
    "ReplicationGroup": {
```

```
        "ReplicationGroupId": "node-test",
        "Description": "node-test"
       },
        "Status": "modifying",
        "PendingModifiedValues": {},
        "MemberClusters": [
            "node-test-001",
            "node-test-002",
            "node-test-003",
            "node-test-004",
            "node-test-005",
            "node-test-006"
        ],
        "NodeGroups": [
            {
                "NodeGroupId": "0001",
                "Status": "modifying",
                "PrimaryEndpoint": {
                    "Address": "node-test.zzzzzz.ng.0001.usw2.cache.amazonaws.com",
                    "Port": 6379
                },
                "ReaderEndpoint": {
                    "Address": "node-test-
ro.zzzzzz.ng.0001.usw2.cache.amazonaws.com",
                    "Port": 6379
                },
                "NodeGroupMembers": [
                    {
                        "CacheClusterId": "node-test-001",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address": "node-
test-001.zzzzzz.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2a",
                        "CurrentRole": "primary"
                    },
                    {
                        "CacheClusterId": "node-test-002",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address": "node-
test-002.zzzzzz.0001.usw2.cache.amazonaws.com",
```

```
                        "Port": 6379
                    },
                    "PreferredAvailabilityZone": "us-west-2c",
                    "CurrentRole": "replica"
                },
                {
                    "CacheClusterId": "node-test-003",
                    "CacheNodeId": "0001",
                    "ReadEndpoint": {
                        "Address": "node-
test-003.zzzzzz.0001.usw2.cache.amazonaws.com",
                        "Port": 6379
                    },
                    "PreferredAvailabilityZone": "us-west-2b",
                    "CurrentRole": "replica"
                },
                {
                    "CacheClusterId": "node-test-004",
                    "CacheNodeId": "0001",
                    "ReadEndpoint": {
                        "Address": "node-
test-004.zzzzzz.0001.usw2.cache.amazonaws.com",
                        "Port": 6379
                    },
                    "PreferredAvailabilityZone": "us-west-2c",
                    "CurrentRole": "replica"
                },
                {
                    "CacheClusterId": "node-test-005",
                    "CacheNodeId": "0001",
                    "ReadEndpoint": {
                        "Address": "node-
test-005.zzzzzz.0001.usw2.cache.amazonaws.com",
                        "Port": 6379
                    },
                    "PreferredAvailabilityZone": "us-west-2b",
                    "CurrentRole": "replica"
                },
                {
                    "CacheClusterId": "node-test-006",
                    "CacheNodeId": "0001",
                    "ReadEndpoint": {
                        "Address": "node-
test-006.zzzzzz.0001.usw2.cache.amazonaws.com",
```

```
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2b",
                        "CurrentRole": "replica"
                    }
                ]
            }
        ],
        "SnapshottingClusterId": "node-test-002",
        "AutomaticFailover": "enabled",
        "MultiAZ": "enabled",
        "SnapshotRetentionLimit": 1,
        "SnapshotWindow": "07:30-08:30",
        "ClusterEnabled": false,
        "CacheNodeType": "cache.r5.large",
         "DataTiering": "disabled",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false,
        "ARN": "arn:aws:elasticache:us-west-2:123456789012:replicationgroup:node-
test"
    }
}
```

También puede llamar a `decrease-replica-count` y en lugar de pasar el parámetro `--new-replica-count`, podría pasar el parámetro `--replicas-to-remove`, como se muestra a continuación:

Para Linux, macOS o Unix:

```
aws elasticache decrease-replica-count \
    --replication-group-id my-replication-group \
    --replicas-to-remove node-test-003 \
    --region us-east-2 \
    --apply-immediately
```

Para Windows:

```
aws elasticache decrease-replica-count ^
    --replication-group-id my-replication-group ^
    --replicas-to-remove node-test-003 ^
    --region us-east-2 ^
```

```
    --apply-immediately
```

Para obtener más información, consulte los temas de la AWS CLI `decrease-replica-count`.

Uso de la API ElastiCache con Valkey o Redis OSS

Para eliminar nodos mediante la API de ElastiCache, llame a la operación de API `DecreaseReplicaCount` con el ID de grupo de reproducción y una lista de nodos a fin de eliminar, como se muestra a continuación:

- `ReplicationGroupId`: el ID del grupo de reproducción del que desea eliminar nodos.
- `ReplicasToRemove` El parámetro `ReplicasToRemove` especifica el número de nodos que desea en este clúster una vez aplicada la modificación.
- `ApplyImmediately` especifica si desea quitar estos nodos de forma inmediata o durante el próximo periodo de mantenimiento.
- `Region` especifica la región de AWS del clúster del que desea quitar un nodo.

En el ejemplo siguiente, se quitan de inmediato los nodos 0004 y 0005 del clúster my-cluster.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=DecreaseReplicaCount
    &ReplicationGroupId=my-replication-group
    &ApplyImmediately=true
    &ReplicasToRemove=node-test-003
    &Region us-east-2
    &Version=2014-12-01
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20141201T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Para obtener más información, consulte el tema de la API de ElastiCache `DecreaseReplicaCount`.

Uso de la API de ElastiCache con Memcached

Para eliminar nodos mediante la API de ElastiCache, llame a la operación de API `ModifyCacheCluster` con el ID de clúster de caché y una lista de nodos a fin de eliminar, como se muestra a continuación:

- `CacheClusterId` El ID del clúster del que desea quitar nodos.
- `NumCacheNodes` El parámetro `NumCacheNodes` especifica el número de nodos que desea en este clúster una vez aplicada la modificación.
- `CacheNodeIdsToRemove.member.n` La lista de los ID de los nodos para quitar del clúster.
  - `CacheNodeIdsToRemove.member.1=0004`
  - `CacheNodeIdsToRemove.member.1=0005`
- `ApplyImmediately` especifica si desea quitar estos nodos de forma inmediata o durante el próximo periodo de mantenimiento.
- `Region` especifica la región de AWS del clúster del que desea quitar un nodo.

En el ejemplo siguiente, se quitan de inmediato los nodos 0004 y 0005 del clúster my-cluster.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=ModifyCacheCluster
    &CacheClusterId=my-cluster
    &ApplyImmediately=true
    &CacheNodeIdsToRemove.member.1=0004
    &CacheNodeIdsToRemove.member.2=0005
    &NumCacheNodes=3
    &Region us-east-2
    &Version=2014-12-01
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20141201T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Para obtener más información, consulte el tema de la API de ElastiCache `ModifyCacheCluster`.

Cancelación de operaciones pendientes de adición o eliminación de nodos en ElastiCache

Si ha optado por no aplicar un cambio de clúster de ElastiCache de forma inmediata, la operación tiene el estado Pendiente hasta que se realice en el próximo periodo de mantenimiento. Puede cancelar cualquier operación de adición o eliminación pendiente.

Para cancelar una operación para añadir o eliminar nodos que haya quedado pendiente en la AWS CLI, utilice el comando `modify-cache-cluster`. Establezca `num-cache-nodes` igual al número actual de nodos de caché del clúster y, a continuación, añada el indicador `--apply-immediately`. Esto anulará el cambio pendiente.

Para cancelar la adición o eliminación de un nodo pendiente:

```
aws elasticache modify-cache-cluster
 --cache-cluster-id <your-cluster-id>
 --num-cache-nodes <current-number-of-nodes>
 --apply-immediately
```

Si no está claro si las adiciones o eliminaciones de nodos están pendientes, puede confirmar su estado con el comando `describe-cache-clusters`:

```
aws elasticache describe-cache-clusters
 --cache-cluster-id <your-cluster-id>
```

Todos los nodos pendientes deberían aparecer en la salida `PendingModifiedValues`. Por ejemplo:

```
"PendingModifiedValues": {
 "NumCacheNodes": 3
 },
```

Eliminación de un clúster en ElastiCache

Siempre que un clúster de ElastiCache tenga el estado Disponible, se cobrará por él, independientemente de si lo está usando de forma activa o no. Para que dejen de devengarse cargos, elimine el clúster.

> **Warning**
>
> Cuando se elimina un clúster de ElastiCache, se retienen las instantáneas manuales. También puede crear una instantánea final antes de eliminar el clúster. Por el contrario, las instantáneas de caché automáticas no se conservan.

Uso de Consola de administración de AWS

El siguiente procedimiento elimina un único clúster de su implementación. Para eliminar varios clústeres, repita el procedimiento por cada clúster que desee eliminar. No es necesario esperar a un clúster para terminar de eliminarlo antes de empezar el procedimiento para eliminar otro clúster.

Para eliminar un clúster

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel del motor de ElastiCache, elija el motor en el que se ejecuta el clúster que desea eliminar.

   Se mostrará una lista de todos los clústeres que ejecutan ese motor.
3. Para elegir el clúster que se va a eliminar, elija el nombre del clúster de la lista de clústeres.

   > **Important**
   >
   > Los clústeres solo se pueden eliminar uno por uno en la consola de ElastiCache. Al elegir varios clústeres, se deshabilita la operación de eliminación.
4. En Actions (Acciones), seleccione Delete (Eliminar).
5. En la pantalla de confirmación Delete Cluster (Eliminar clúster), elija Delete (Eliminar) para eliminar el clúster o elija Cancel (Cancelar) para conservarlo.

   Si elige Delete, el estado del clúster cambia a deleting.

En cuanto el clúster desaparezca de la lista de clústeres, dejará de incurrir en gastos.

Uso de AWS CLI para eliminar un clúster de ElastiCache

El código siguiente elimina el clúster de ElastiCache `my-cluster`.

```
aws elasticache delete-cache-cluster --cache-cluster-id my-cluster
```

La acción `delete-cache-cluster` de la CLI solo elimina un clúster. Para eliminar varios clústeres, llame a `delete-cache-cluster` por cada clúster de caché que desee eliminar. No es necesario esperar a que se termine de eliminar un clúster antes de eliminar otro.

Para Linux, macOS o Unix:

```
aws elasticache delete-cache-cluster \
    --cache-cluster-id my-cluster \
    --region us-east-2
```

Para Windows:

```
aws elasticache delete-cache-cluster ^
    --cache-cluster-id my-cluster ^
    --region us-east-2
```

Para obtener más información, consulte la AWS CLI para el tema de ElastiCache `delete-cache-cluster`.

Uso de la API de ElastiCache

El código siguiente elimina el clúster `my-cluster`.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=DeleteCacheCluster
    &CacheClusterId=my-cluster
    &Region us-east-2
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T220302Z
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20150202T220302Z
    &X-Amz-SignedHeaders=Host
```

```
    &X-Amz-Expires=20150202T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

La operación de API `DeleteCacheCluster` solo elimina un clúster. Para eliminar varios clústeres, llame a `DeleteCacheCluster` por cada clúster que desee eliminar. No es necesario esperar a que se termine de eliminar un clúster antes de eliminar otro.

Para obtener más información, consulte el tema de referencia de la API de ElastiCache `DeleteCacheCluster`.

Acceso al clúster o grupo de replicación de ElastiCache

Las instancias de Amazon ElastiCache se han diseñado para acceder a ellas a través de una instancia de Amazon EC2.

Si ha lanzado la instancia de ElastiCache en una Amazon Virtual Private Cloud (Amazon VPC), puede acceder a la instancia de ElastiCache desde una instancia de Amazon EC2 en la misma Amazon VPC. O, mediante la utilización de la conexión de emparejamiento de VPC, puede acceder a la instancia de ElastiCache desde una instancia de Amazon EC2 en otra Amazon VPC.

Si ha lanzado la instancia de ElastiCache en EC2 Classic, debe permitir que la instancia EC2 obtenga acceso a su clúster; para ello, conceda al grupo de seguridad de Amazon EC2 asociado a la instancia acceso a su grupo de seguridad de caché. De forma predeterminada, el acceso a un clúster está restringido a la cuenta que lanzó el clúster.

Temas

- Conceder acceso al clúster o al grupo de reproducción

Conceder acceso al clúster o al grupo de reproducción

Ha lanzado su clúster en EC2-VPC

Si ha lanzado el clúster en una Amazon Virtual Private Cloud (Amazon VPC), puede conectarse al clúster de ElastiCache solo desde una instancia de Amazon EC2 que se ejecuta en la misma Amazon VPC. En este caso, necesitará conceder acceso de red al clúster.

> **Note**
>
> Si utiliza Local Zones, asegúrese de haberlo habilitado. Para obtener más información, consulte Habilitar Local Zones. Al hacerlo, la VPC se extiende a esa zona local y tratará la subred como cualquier subred en cualquier otra zona de disponibilidad. Las gateways relevantes, las tablas de enrutamiento y otras consideraciones del grupo de seguridad se ajustarán de forma automática.

Para conceder acceso de red desde un grupo de seguridad de Amazon VPC a un clúster

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon EC2 en https://console.aws.amazon.com/ec2/.

2. En el panel de navegación, en Network & Security (Redes y seguridad), seleccione Security Groups(Grupos de seguridad).
3. En la lista de grupos de seguridad, elija el grupo de seguridad para su Amazon VPC. A menos que haya creado un grupo de seguridad para el uso de ElastiCache, este grupo de seguridad se denominará default (predeterminado).
4. Elija la pestaña Inbound y haga lo siguiente:

    a. Elija Editar.

    b. Seleccione Agregar regla.

    c. En la columna Type, elija Custom TCP rule.

    d. En el cuadro Port range, escriba el número de puerto para su nodo de clúster. Este número debe ser el mismo que especificó cuando lanzó el clúster. El puerto predeterminado para Memcached es **11211**. El puerto predeterminado para Valkey y Redis OSS es **6379**.

    e. En el cuadro de Source (Fuente), elija Anywhere (En cualquier lugar) que tenga el rango de puertos (0.0.0.0/0) para que cualquier instancia de Amazon EC2 que lance en su Amazon VPC pueda conectarse a sus nodos de ElastiCache.

    > **Important**
    >
    > Al abrir el clúster de ElastiCache a 0.0.0.0/0 no se expone el clúster a Internet, ya que no tiene una dirección IP pública y, por lo tanto, no se puede acceder a este desde fuera de la VPC. Sin embargo, el grupo de seguridad predeterminado se puede aplicar a otras instancias de Amazon EC2 en la cuenta del cliente y dichas instancias pueden tener una dirección IP pública. Si se está ejecutando algo en el puerto predeterminado, ese servicio podría exponerse de forma involuntaria. Por lo tanto, recomendamos crear un grupo de seguridad de VPC que ElastiCache utilice de forma exclusiva. Para obtener más información, consulte Grupos de seguridad personalizados.

    f. Seleccione Save (Guardar).

Al lanzar una instancia de Amazon EC2 en la Amazon VPC, esa instancia podrá conectarse al clúster de ElastiCache.

Acceso a recursos de ElastiCache desde fuera de AWS

Amazon ElastiCache es un servicio de AWS con un almacén de valores de claves en memoria basado en la nube. El servicio está diseñado para acceder solo desde AWS. Sin embargo, si el clúster de ElastiCache está alojado en una VPC, se puede usar una instancia de Traducción de direcciones de red (NAT) para proporcionar acceso externo.

Requisitos

Para poder obtener acceso a los recursos de ElastiCache desde fuera de AWS, deben cumplirse los siguientes requisitos:

- El clúster debe estar en una VPC y el acceso debe hacerse a través de una instancia de Traducción de direcciones de red (NAT). Este requisito no tiene excepciones.
- La instancia NAT debe lanzarse en la misma VPC que el clúster.
- La instancia NAT debe lanzarse en una subred pública distinta de la del clúster.
- Es necesario asociar una dirección IP elástica (EIP) a la instancia NAT. La característica de reenvío de puertos de las tablas IP se usa para reenviar un puerto de la instancia NAT al puerto del nodo de caché de la VPC.

Consideraciones

Tenga en cuenta los siguientes factores a la hora de acceder a sus recursos de ElastiCache desde fuera de ElastiCache.

- Los clientes se conectan a la EIP y al puerto de caché de la instancia NAT. El reenvío de puertos de la instancia NAT reenvía el tráfico al nodo de clúster adecuado.
- Si un nodo del clúster se añade o se reemplaza, las reglas de tablas IP deben actualizarse para reflejar este cambio.

Limitaciones

Este enfoque solo debe usarse con fines de prueba y desarrollo. No se recomienda su uso para la producción debido a las limitaciones siguientes:

- La instancia NAT actúa como proxy entre los clientes y múltiples clústeres. La adición de un proxy afecta al rendimiento del clúster. El impacto aumenta con el número de clústeres a los que se accede mediante la instancia NAT.

- El tráfico de los clientes a la instancia NAT está sin cifrar. Por lo tanto, debe evitar el envío de información confidencial a través de la instancia NAT.
- La instancia NAT requiere, además, el mantenimiento de otra instancia.
- La instancia NAT sirve como único punto de error. Para obtener información acerca de cómo configurar NAT de alta disponibilidad en una VPC, consulte Alta disponibilidad para instancias NAT de Amazon VPC: un ejemplo.

Cómo acceder a recursos de ElastiCache desde fuera de AWS

El siguiente procedimiento muestra cómo conectarse a sus recursos de ElastiCache mediante una instancia NAT.

En los pasos que se describen a continuación se da por sentado lo siguiente:

- `iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6380 -j DNAT --to 10.0.1.231:6379`
- `iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6381 -j DNAT --to 10.0.1.232:6379`

A continuación, necesita NAT en la dirección opuesta:

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to-source 10.0.0.55
```

También debe activar el reenvío de IP, que está desactivado de forma predeterminada:

```
sudo sed -i 's/net.ipv4.ip_forward=0/net.ipv4.ip_forward=1/g' /etc/
sysctl.conf sudo sysctl --system
```

- Está accediendo a un clúster de Memcached con:
  - Dirección IP: 10.0.1.230
  - Puerto de Memcached predeterminado: 11 211
  - Grupo de seguridad: *10\.0\.0\.55*
- Está accediendo a un clúster de Valkey o Redis OSS con:
  - Dirección IP: 10.0.1.230
  - Puerto predeterminado: 6379
  - Grupo de seguridad: sg-bd56b7da

  - Dirección IP de la instancia de AWS: 198.99.100.27
- El cliente tiene de confianza tiene la dirección IP 198.51.100.27.
- La instancia NAT tiene la dirección IP elástica 203.0.113.73.
- La instancia NAT tiene el grupo de seguridad sg-ce56b7a9.

Para conectarse a sus recursos de ElastiCache mediante una instancia NAT

1. Cree una instancia NAT en la misma VPC que su clúster, pero en una subred pública.

   De forma predeterminada, el asistente de VPC lanzara el tipo de nodo cache.m1.small. Seleccione un tamaño de nodo en función de sus necesidades. A fin de obtener acceso a ElastiCache desde fuera de AWS, debe utilizar una EC2 con una AMI con NAT.

   Para obtener información acerca de la creación de instancias NAT, consulte Instancias NAT en la Guía del usuario de AWS VPC.
2. Cree reglas de grupo de seguridad para el clúster y la instancia NAT.

   El grupo de seguridad de la instancia NAT y la instancia del clúster deben tener las siguientes reglas:

   - Dos reglas de entrada
     - Con Memcached, la primera regla es permitir las conexiones de TCP desde los clientes de confianza a cada puerto de caché que se reenvía desde la instancia de NAT (11211 - 11213).
     - Con Valkey y Redis OSS, la primera regla es permitir las conexiones de TCP desde los clientes de confianza a cada puerto de caché que se reenvía desde la instancia de NAT (6379 - 6381).
     - Una segunda regla es permitir el acceso de SSH a los clientes de confianza.

   Grupo de seguridad de instancias de NAT: reglas de entrada con Memcached

Tipo	Protocolo	Intervalo de puertos	Origen
Regla TCP personalizada	TCP	11211-11213	198.51.100.27/32

Tipo	Protocolo	Intervalo de puertos	Origen
SSH	TCP	22	198.51.100.27/32

Grupo de seguridad de instancias de NAT: reglas de entrada con Valkey o Redis OSS

Tipo	Protocolo	Intervalo de puertos	Origen
Regla TCP personalizada	TCP	6379-6380	198.51.100.27/32
SSH	TCP	22	203.0.113.73/32

- Con Memcached, una regla de salida para permitir las conexiones de TCP con el puerto de caché (11211).

  Grupo de seguridad de la instancia NAT: regla saliente

Tipo	Protocolo	Rango de puerto	Destino
Regla TCP personalizada	TCP	11211	sg-ce56b7a9 (grupo de seguridad de NAT)

- Con Valkey o Redis OSS, una regla de salida que permite las conexiones de TCP con el puerto de caché (6379).

  Grupo de seguridad de la instancia NAT: regla saliente

Tipo	Protocolo	Rango de puerto	Destino
Regla TCP personalizada	TCP	6379	sg-ce56b7a9 (grupo de seguridad de NAT)

- Con Memcached, una regla de entrada del grupo de seguridad del clúster que permite conexiones de TCP entre la instancia de NAT y el puerto de caché (11211).

Grupo de seguridad de la instancia del clúster: regla entrante

Tipo	Protocolo	Intervalo de puertos	Origen
Regla TCP personalizada	TCP	11211	sg-ce56b7 a9 (grupo de seguridad de NAT)

- Con Valkey o Redis OSS, una regla de entrada del grupo de seguridad del clúster que permite conexiones de TCP entre la instancia de NAT y el puerto de caché (6379).

  Grupo de seguridad de la instancia del clúster: regla entrante

Tipo	Protocolo	Intervalo de puertos	Origen
Regla TCP personalizada	TCP	6379	sg-ce56b7 a9 (grupo de seguridad de NAT)

3. Valide las reglas.

   - Confirme que el cliente de confianza es capaz de conectarse mediante SSH a la instancia NAT.
   - Confirme que el cliente de confianza es capaz de conectarse al clúster desde la instancia NAT.

4. Memcached

   Añada una regla de tablas IP a la instancia NAT.

   La regla de tablas de IP deben añadirse a la tabla NAT para que todos los nodos del clúster reenvíen el puerto de caché de la instancia NAT al nodo del clúster. Un ejemplo de esto sería el siguiente:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11211 -j DNAT --to
 10.0.1.230:11211
```

   El número de puerto debe ser único para cada nodo del clúster. Por ejemplo, si trabaja con un clúster de Memcached de tres nodos que usan los puertos 11211 - 11213, las reglas deben ser similares a las que se muestran a continuación:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11211 -j DNAT --to
 10.0.1.230:11211
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11212 -j DNAT --to
 10.0.1.231:11211
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11213 -j DNAT --to
 10.0.1.232:11211
```

Confirme que el cliente de confianza es capaz de conectarse al clúster.

El cliente de confianza deben conectarse a la EIP asociada con la instancia NAT y el puerto del clúster correspondiente al nodo de clúster adecuado. Por ejemplo, la cadena de conexión para PHP puede ser parecida a la siguiente:

```
$memcached->connect( '203.0.113.73', 11211 );
$memcached->connect( '203.0.113.73', 11212 );
$memcached->connect( '203.0.113.73', 11213 );
```

También puede usar un cliente de Telnet para comprobar la conexión. Por ejemplo:

```
telnet 203.0.113.73 11211
telnet 203.0.113.73 11212
telnet 203.0.113.73 11213
```

Valkey o Redis OSS

Añada una regla de tablas IP a la instancia NAT.

La regla de tablas de IP deben añadirse a la tabla NAT para que todos los nodos del clúster reenvíen el puerto de caché de la instancia NAT al nodo del clúster. Un ejemplo de esto sería el siguiente:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6379 -j DNAT --to
 10.0.1.230:6379
```

El número de puerto debe ser único para cada nodo del clúster. Por ejemplo, si trabaja con un clúster de Redis OSS de tres nodos que utiliza los puertos 6379 - 6381, las reglas serían similares a las que se muestran a continuación:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6379 -j DNAT --to
 10.0.1.230:6379
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6380 -j DNAT --to
 10.0.1.231:6379
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6381 -j DNAT --to
 10.0.1.232:6379
```

Confirme que el cliente de confianza es capaz de conectarse al clúster.

El cliente de confianza deben conectarse a la EIP asociada con la instancia NAT y el puerto del clúster correspondiente al nodo de clúster adecuado. Por ejemplo, la cadena de conexión para PHP puede ser parecida a la siguiente:

```
redis->connect( '203.0.113.73', 6379 );
redis->connect( '203.0.113.73', 6380 );
redis->connect( '203.0.113.73', 6381 );
```

También puede usar un cliente de Telnet para comprobar la conexión. Por ejemplo:

```
telnet 203.0.113.73 6379
telnet 203.0.113.73 6380
telnet 203.0.113.73 6381
```

5. Guarde la configuración de tablas IP.

   Guarde las reglas tras realizar las pruebas y verificarlas. Si usa una distribución de Linux basada en Redhat (como Amazon Linux) ejecute el comando siguiente:

   ```
   service iptables save
   ```

Temas relacionados

Los temas siguientes pueden ser de su interés.

- Patrones de acceso para obtener acceso a una caché de ElastiCache en una Amazon VPC
- Acceso a una caché de ElastiCache desde una aplicación en ejecución en un centro de datos del cliente
- Instancias NAT

- Configuración de clientes de ElastiCache
- Alta disponibilidad para instancias NAT de Amazon VPC: un ejemplo

Búsqueda de puntos de conexión en ElastiCache

Su aplicación se conecta a su clúster de ElastiCache mediante puntos de conexión. Un punto de conexión es un nodo o la dirección única del clúster.

Puede establecer una conexión privada entre los puntos de conexión de la VPC y la API de ElastiCache mediante la creación de un punto de conexión de VPC de interfaz desde AWS PrivateLink. Para obtener más información, consulte Puntos de conexión de VPC de interfaz y API de ElastiCache (AWS PrivateLink).

Puntos de conexión que se deben usar con Valkey o Redis OSS.

- Para el Nodo independiente, use el punto de conexión del nodo para las operaciones de lectura y escritura.

- Para Clústeres de Valkey o Redis OSS (modo de clúster desactivado), utilice el Punto de conexión principal para todas las operaciones de escritura. Use el punto de conexión de lector para dividir de manera uniforme las conexiones entrantes al punto de enlace entre todas las réplicas de lectura. Use los puntos de conexión del nodo para las operaciones de lectura (en la API/CLI, estas se denominan puntos de conexión de lectura).

- Para Clústeres de Valkey o Redis OSS (modo de clúster activado), utilice el Punto de conexión de configuración del clúster para todas las operaciones que admiten comandos del modo de clúster activado. Debe usar un cliente que admita clústeres de Valkey o Redis OSS en Redis OSS 3.2 y versiones posteriores. También podrá realizar operaciones de lectura desde puntos de conexión de nodos individuales (en la API/CLI, estas se denominan puntos de conexión de lectura).

Las secciones siguientes le guiarán en el proceso de detección de los puntos de conexión que necesitará para el motor que esté usando.

Puntos de conexión que se deben usar con Memcached.

Para Caché sin servidor de ElastiCache con Memcached, basta con adquirir el DNS del punto de conexión del clúster y el puerto desde la consola.

Desde AWS CLI, utilice el comando `describe-serverless-caches` para adquirir información sobre el punto de conexión.

Linux

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

La salida de la operación anterior tendrá un aspecto similar al siguiente (formato JSON):

```
{
    "ServerlessCaches": [
        {
            "ServerlessCacheName": "serverless-memcached",
            "Description": "test",
            "CreateTime": 1697659642.136,
            "Status": "available",
            "Engine": "memcached",
            "MajorEngineVersion": "1.6",
            "FullEngineVersion": "21",
            "SecurityGroupIds": [
                "sg-083eda453e1e51310"
            ],
            "Endpoint": {
                "Address": "serverless-memcached-01.amazonaws.com",
                "Port":11211
            },
            "ARN": "<the ARN>",
            "SubnetIds": [
                "subnet-0cf759df15bd4dc65",
                "subnet-09e1307e8f1560d17"
            ],
            "SnapshotRetentionLimit": 0,
            "DailySnapshotTime": "03:00"
        }
    ]
```

```
}
```

Para un clúster de Memcached basado en instancias, si usa la función de detección automática, puede usar el punto de conexión de configuración del clúster para configurar el cliente de Memcached. Esto significa que debe usar un cliente que admita la función de detección automática.

Si opta por no usar la función de detección automática, deberá configurar su cliente para usar los puntos de conexión individuales para las operaciones de lectura y escritura. También deberá realizar un seguimiento conforme agregue y elimine nodos.

Búsqueda de puntos de conexión de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) (consola)

Si un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) solo tiene un nodo, el punto de conexión de dicho nodo se utiliza tanto para las operaciones de lectura como de escritura. Si un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) tiene varios nodos, hay tres tipos de puntos de conexión: el punto de conexión principal, el punto de conexión del lector y los puntos de conexión de nodo.

El punto de conexión principal es un nombre de DNS que siempre se resuelve en el nodo principal del clúster. El punto de conexión principal es inmune a los cambios en el clúster, como la promoción de una réplica de lectura a la función principal. Para una actividad de escritura, recomendamos que sus aplicaciones se conecten al punto de conexión principal.

Un punto de conexión del lector dividirá de manera uniforme las conexiones entrantes al punto de conexión entre todas las réplicas de lectura de un clúster de ElastiCache para Redis OSS. Factores adicionales como cuando la aplicación crea las conexiones o cómo la aplicación (re)utiliza las conexiones determinará la distribución del tráfico. Los puntos de enlace de lector se mantienen actualizados con los cambios del clúster en tiempo real cuando se añaden o eliminan réplicas. Puede colocar las diferentes réplicas de lectura del clúster de ElastiCache para Redis OSS en distintas zonas de disponibilidad (AZ) de AWS para garantizar la alta disponibilidad de los puntos de conexión del lector.

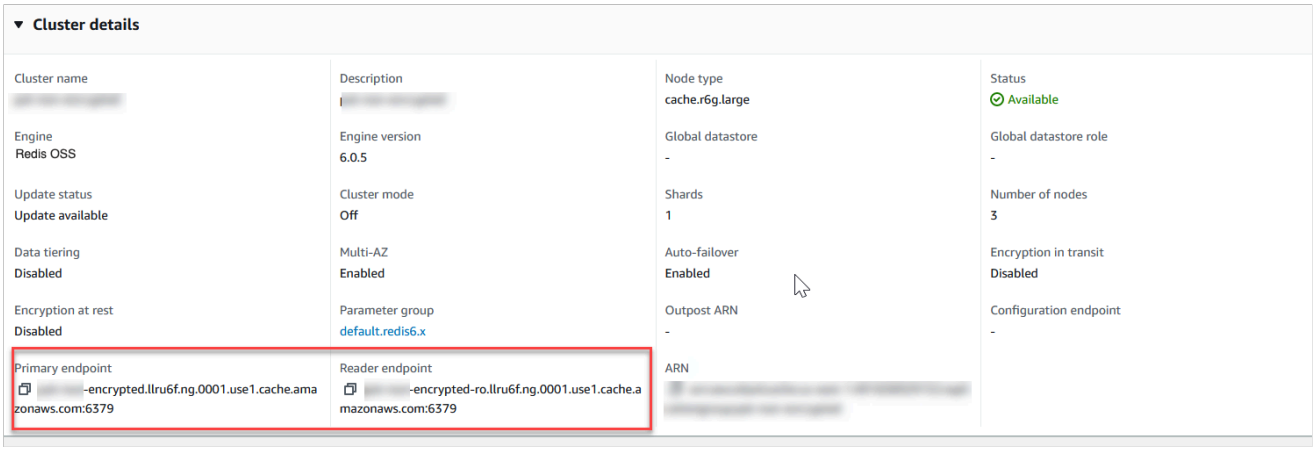
> **Note**
>
> Un punto de conexión del lector no es un equilibrador de carga. Se trata de un registro de DNS que se resolverá en una dirección IP de uno de los nodos de réplica en forma de turno rotativo.

Para la actividad de lectura, las aplicaciones pueden conectarse también a cualquier nodo del clúster. A diferencia del punto de conexión principal, los puntos de conexión de nodo se resuelven en puntos de enlace específicos. Si realiza un cambio en su clúster, como añadir o eliminar una réplica, debe actualizar los puntos de enlace del nodo en su aplicación.

Búsqueda de puntos de conexión de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Clústeres de Valkey o Clústeres de Redis OSS.

   Aparecerá la pantalla de clústeres con una lista de clústeres de Valkey o Redis OSS (modo de clúster deshabilitado) y de clústeres de Valkey o Redis OSS (modo de clúster habilitado).
3. Para buscar los puntos de conexión principal o de lector, elija el nombre del clúster (no el botón a su izquierda).

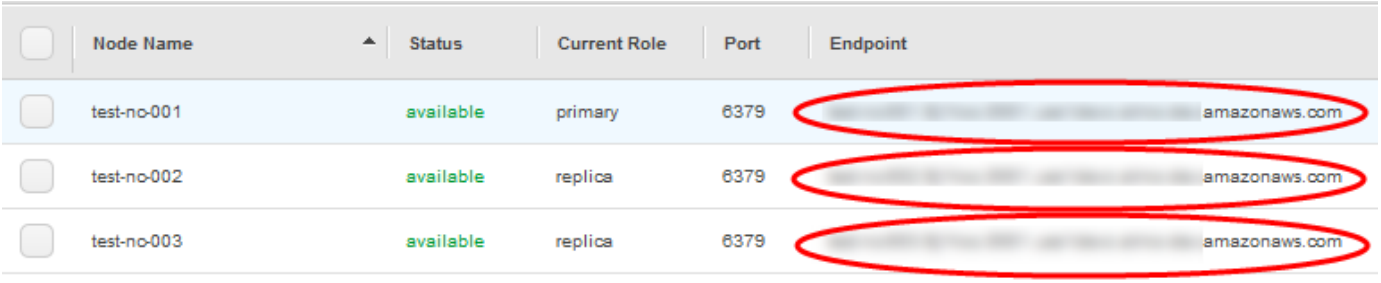


Puntos de conexión del lector y principal para un clúster de Valkey o Redis OSS (modo de clúster deshabilitado)

Si el clúster tiene solo un nodo, quiere decir que no hay punto de conexión principal y, por lo tanto, puede continuar con el paso siguiente.

4. Si el clúster de Valkey o Redis OSS (modo de clúster deshabilitado) tiene nodos de réplica, puede encontrar los puntos de conexión del nodo de réplica del clúster eligiendo el nombre del clúster y, a continuación, la pestaña Nodos.

   Aparecerá la pantalla de nodos con los nodos del clúster, principal y réplicas; cada uno con su punto de conexión.



Puntos de conexión del nodo para un clúster de Valkey o Redis OSS (modo de clúster deshabilitado)

5. Para copiar un punto de conexión al portapapeles:

   a. Un punto de conexión cada vez, encuentre el punto de conexión que desea copiar.

   b. Elija el icono de copia directamente delante del punto de conexión.

   El punto de conexión se ha copiado ahora en el portapapeles. Para obtener información sobre el uso del punto de conexión a fin de conectarse a un nodo, consulte Conexión a los nodos de Memcached.

Un punto de conexión principal de Valkey o Redis OSS (modo de clúster deshabilitado) tiene un aspecto parecido al siguiente. Existe una diferencia dependiendo de si está habilitado el cifrado en tránsito o no lo está.

Cifrado en tránsito no habilitado

```
clusterName.xxxxxx.nodeId.regionAndAz.cache.amazonaws.com:port

redis-01.7abc2d.0001.usw2.cache.amazonaws.com:6379
```

Cifrado en tránsito habilitado

```
master.clusterName.xxxxxx.regionAndAz.cache.amazonaws.com:port

master.ncit.ameaqx.use1.cache.amazonaws.com:6379
```

Búsqueda de puntos de conexión para un clúster de Valkey o Redis OSS (modo de clúster habilitado) (consola)

Un clúster de Valkey o Redis OSS (modo de clúster habilitado) tiene un único punto de conexión de configuración. Al conectarse al punto de conexión de configuración, la aplicación puede detectar los puntos de conexión principal y de lectura de cada partición del clúster.

Búsqueda de puntos de conexión del clúster de Valkey o Redis OSS (modo de clúster habilitado)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Clústeres de Valkey o Clústeres de Redis OSS.

   Aparecerá la pantalla de clústeres con una lista de clústeres. Haga clic en el clúster al que desea conectarse.
3. Para buscar el punto de conexión de configuración del clúster, elija el nombre del clúster (no el botón de opción).
4. El Configuration endpoint (Punto de conexión de configuración) se muestra en Cluster details(Detalles de clúster). Para copiarlo, elija el ícono copiar a la izquierda del punto de conexión.

Búsqueda de puntos de conexión de un clúster (consola) (Memcached)

Todos los puntos de conexión de Memcached son de lectura/escritura. Para conectarse a los nodos de un clúster de Memcached, su aplicación puede usar los puntos de conexión de cada nodo o el punto de conexión de configuración del clúster con la función de detección automática. Para usar la función de detección automática, debe usar un cliente que admita la función de detección automática.

Cuando se usa la función de detección automática, la aplicación cliente se conecta a su clúster de Memcached mediante el punto de conexión de configuración. A medida que amplíe su clúster agregando o quitando nodos, su aplicación "sabrá" automáticamente los nodos que hay en el clúster y se podrá conectar a cualquiera de ellos. Sin la función de detección automática, su aplicación tendría que averiguar los nodos existentes o tendría que actualizar manualmente los puntos de conexión de su aplicación cada vez que se agrega o quita un nodo.

Para copiar un punto de conexión, elija el icono de copia situado justo delante de la dirección del punto de conexión. Para obtener información sobre el uso del punto de conexión a fin de conectarse a un nodo, consulte Conexión a los nodos de Memcached.

Los puntos de conexión de configuración y del nodo son muy similares. A continuación se muestran las diferencias resaltadas en negrita.

```
myclustername.xxxxxx.cfg.usw2.cache.amazonaws.com:port   # configuration endpoint
 contains "cfg"
myclustername.xxxxxx.0001.usw2.cache.amazonaws.com:port  # node endpoint for node 0001
```

> **Important**
>
> Si decide crear un CNAME para su punto de conexión de Memcached, para que el cliente de detección automática reconozca el CNAME como un punto de conexión de configuración, deberá incluir `.cfg.` en el CNAME.

Búsqueda de puntos de conexión (AWS CLI)

Para Memcached, puede utilizar la AWS CLI de Amazon ElastiCache para detectar los puntos de conexión de los nodos y los clústeres.

Para Redis OSS, puede utilizar la AWS CLI de Amazon ElastiCache para detectar los puntos de conexión de los nodos, los clústeres y los grupos de replicación.

Temas

- Búsqueda de puntos de conexión para nodos y clústeres (AWS CLI)
- Búsqueda de los puntos de conexión de los grupos de replicación de Valkey o Redis OSS (AWS CLI)

Búsqueda de puntos de conexión para nodos y clústeres (AWS CLI)

Puede usar AWS CLI para detectar los puntos de enlace de un clúster y sus nodos con el comando `describe-cache-clusters`. Para clústeres de Valkey o Redis OSS, el comando devuelve el punto de conexión del clúster. Para clústeres de Memcached, el comando devuelve el punto de conexión de configuración. Si incluye el parámetro opcional `--show-cache-node-info`, el comando también devolverá los puntos de conexión de los nodos individuales del clúster.

Example

El comando siguiente recupera el punto de conexión de configuración (`ConfigurationEndpoint`) y los puntos de conexión de nodos individuales (`Endpoint`) para el clúster de Memcached mycluster.

Para Linux, macOS o Unix:

```
aws elasticache describe-cache-clusters \
    --cache-cluster-id mycluster \
    --show-cache-node-info
```

Para Windows:

```
aws elasticache describe-cache-clusters ^
    --cache-cluster-id mycluster ^
    --show-cache-node-info
```

La salida de la operación anterior tendrá un aspecto similar al siguiente (formato JSON).

```
{
   "CacheClusters": [
   {
       "Engine": "memcached",
       "CacheNodes": [
          {
             "CacheNodeId": "0001",
             "Endpoint": {
                "Port": 11211,
                "Address": "mycluster.amazonaws.com"
             },
                "CacheNodeStatus": "available",
                "ParameterGroupStatus": "in-sync",
                "CacheNodeCreateTime": "2016-09-22T21:30:29.967Z",
                "CustomerAvailabilityZone": "us-west-2b"
          },
          {
             "CacheNodeId": "0002",
             "Endpoint": {
                "Port": 11211,
                "Address": "mycluster.amazonaws.com"
             },
                "CacheNodeStatus": "available",
                "ParameterGroupStatus": "in-sync",
                "CacheNodeCreateTime": "2016-09-22T21:30:29.967Z",
                "CustomerAvailabilityZone": "us-west-2b"
          },
          {
                "CacheNodeId": "0003",
                "Endpoint": {
                   "Port": 11211,
                   "Address": "mycluster.amazonaws.com"
                },
                   "CacheNodeStatus": "available",
                   "ParameterGroupStatus": "in-sync",
                   "CacheNodeCreateTime": "2016-09-22T21:30:29.967Z",
                   "CustomerAvailabilityZone": "us-west-2b"
          }
       ],
       "CacheParameterGroup": {
       "CacheNodeIdsToReboot": [],
       "CacheParameterGroupName": "default.memcached1.4",
```

```
        "ParameterApplyStatus": "in-sync"
             },
             "CacheClusterId": "mycluster",
             "PreferredAvailabilityZone": "us-west-2b",
             "ConfigurationEndpoint": {
                 "Port": 11211,
                 "Address": "mycluster.amazonaws.com"
             },
             "CacheSecurityGroups": [],
             "CacheClusterCreateTime": "2016-09-22T21:30:29.967Z",
             "AutoMinorVersionUpgrade": true,
             "CacheClusterStatus": "available",
             "NumCacheNodes": 3,
             "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
             "CacheSubnetGroupName": "default",
             "EngineVersion": "1.4.24",
             "PendingModifiedValues": {},
             "PreferredMaintenanceWindow": "mon:09:00-mon:10:00",
             "CacheNodeType": "cache.m4.large",
              "DataTiering": "disabled"
         }
    ]
}
```

> **Important**
>
> Si decide crear un CNAME para su punto de conexión de Memcached, a fin de que el cliente de detección automática reconozca el CNAME como un punto de conexión de configuración, deberá incluir `.cfg.` en el CNAME. Por ejemplo, incluya `mycluster.cfg.local` en el archivo php.ini en el parámetro `session.save_path`.

Example

Para Valkey o Redis OSS, el siguiente comando recupera información del clúster para el clúster de nodo único mycluster.

> **Important**
>
> El parámetro `--cache-cluster-id` se puede utilizar con el ID de clúster de Valkey o Redis OSS (modo de clúster deshabilitado) con un solo nodo o con los ID de un nodo

específico de los grupos de replicación. El valor de `--cache-cluster-id` de un grupo de replicación es un valor de 4 dígitos; por ejemplo, `0001`. Si `--cache-cluster-id` es la identificación de un clúster (nodo) de un grupo de replicación, `replication-group-id` se incluye en la salida.

Para Linux, macOS o Unix:

```
aws elasticache describe-cache-clusters \
    --cache-cluster-id redis-cluster \
    --show-cache-node-info
```

Para Windows:

```
aws elasticache describe-cache-clusters ^
    --cache-cluster-id redis-cluster ^
    --show-cache-node-info
```

La salida de la operación anterior tendrá un aspecto similar al siguiente (formato JSON).

```
{
    "CacheClusters": [
        {
            "CacheClusterStatus": "available",
            "SecurityGroups": [
                {
                    "SecurityGroupId": "sg-77186e0d",
                    "Status": "active"
                }
            ],
            "CacheNodes": [
                {
                    "CustomerAvailabilityZone": "us-east-1b",
                    "CacheNodeCreateTime": "2018-04-25T18:19:28.241Z",
                    "CacheNodeStatus": "available",
                    "CacheNodeId": "0001",
                    "Endpoint": {
                        "Address": "redis-cluster.amazonaws.com",
                        "Port": 6379
                    },
                    "ParameterGroupStatus": "in-sync"
```

```
                }
            ],
            "AtRestEncryptionEnabled": false,
            "CacheClusterId": "redis-cluster",
            "TransitEncryptionEnabled": false,
            "CacheParameterGroup": {
                "ParameterApplyStatus": "in-sync",
                "CacheNodeIdsToReboot": [],
                "CacheParameterGroupName": "default.redis3.2"
            },
            "NumCacheNodes": 1,
            "PreferredAvailabilityZone": "us-east-1b",
            "AutoMinorVersionUpgrade": true,
            "Engine": "redis",
            "AuthTokenEnabled": false,
            "PendingModifiedValues": {},
            "PreferredMaintenanceWindow": "tue:08:30-tue:09:30",
            "CacheSecurityGroups": [],
            "CacheSubnetGroupName": "default",
            "CacheNodeType": "cache.t2.small",
             "DataTiering": "disabled"
            "EngineVersion": "3.2.10",
            "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
            "CacheClusterCreateTime": "2018-04-25T18:19:28.241Z"
        }
    ]
}
```

Para obtener más información, consulte el tema describe-cache-clusters.

Búsqueda de los puntos de conexión de los grupos de replicación de Valkey o Redis OSS (AWS CLI)

Puede usar la AWS CLI para detectar los puntos de enlace de un grupo de reproducción y sus clústeres con el comando `describe-replication-groups`. El comando devuelve el punto de conexión principal del grupo de reproducción y una lista de todos los clústeres (nodos) del grupo de reproducción con sus puntos de conexión, junto con el punto de conexión de lector.

La operación siguiente recupera el punto de conexión principal y el punto de conexión de lector para el grupo de reproducción `myreplgroup`. Use el punto de conexión principal para todas las operaciones de escritura.

```
aws elasticache describe-replication-groups \
```

```
    --replication-group-id myreplgroup
```

Para Windows:

```
aws elasticache describe-replication-groups ^
    --replication-group-id myreplgroup
```

La salida de esta operación anterior tendrá un aspecto similar al siguiente (formato JSON).

```
{
   "ReplicationGroups": [
     {
       "Status": "available",
       "Description": "test",
       "NodeGroups": [
         {
            "Status": "available",
               "NodeGroupMembers": [
                  {
                     "CurrentRole": "primary",
                     "PreferredAvailabilityZone": "us-west-2a",
                     "CacheNodeId": "0001",
                     "ReadEndpoint": {
                        "Port": 6379,
                        "Address": "myreplgroup-001.amazonaws.com"
                     },
                     "CacheClusterId": "myreplgroup-001"
                  },
                  {
                     "CurrentRole": "replica",
                     "PreferredAvailabilityZone": "us-west-2b",
                     "CacheNodeId": "0001",
                     "ReadEndpoint": {
                        "Port": 6379,
                        "Address": "myreplgroup-002.amazonaws.com"
                     },
                     "CacheClusterId": "myreplgroup-002"
                  },
                  {
                     "CurrentRole": "replica",
                     "PreferredAvailabilityZone": "us-west-2c",
                     "CacheNodeId": "0001",
                     "ReadEndpoint": {
```

```
                        "Port": 6379,
                        "Address": "myreplgroup-003.amazonaws.com"
                     },
                     "CacheClusterId": "myreplgroup-003"
                  }
               ],
               "NodeGroupId": "0001",
               "PrimaryEndpoint": {
                  "Port": 6379,
                  "Address": "myreplgroup.amazonaws.com"
               },
               "ReaderEndpoint": {
                  "Port": 6379,
                  "Address": "myreplgroup-ro.amazonaws.com"
               }
            }
         ],
         "ReplicationGroupId": "myreplgroup",
         "AutomaticFailover": "enabled",
         "SnapshottingClusterId": "myreplgroup-002",
         "MemberClusters": [
            "myreplgroup-001",
            "myreplgroup-002",
            "myreplgroup-003"
         ],
         "PendingModifiedValues": {}
      }
   ]
}
```

Para obtener más información, consulte describe-replication-groups en la Referencia de los comandos de la AWS CLI.

Búsqueda de puntos de enlace (API de ElastiCache)

Para Memcached, puede utilizar la API de Amazon ElastiCache para detectar los puntos de conexión de los nodos y los clústeres.

Para Redis OSS, puede utilizar la API de Amazon ElastiCache para detectar los puntos de conexión de los nodos, los clústeres y los grupos de replicación.

Temas

- Búsqueda de puntos de enlace para nodos y clústeres (API de ElastiCache)
- Búsqueda de puntos de conexión de grupos de replicación de Valkey o Redis OSS (API de ElastiCache)

Búsqueda de puntos de enlace para nodos y clústeres (API de ElastiCache)

Puede utilizar la API de ElastiCache para detectar los puntos de enlace de un clúster y sus nodos con la acción `DescribeCacheClusters`. Para clústeres de Valkey o Redis OSS, el comando devuelve el punto de conexión del clúster. Para clústeres de Memcached, el comando devuelve el punto de conexión de configuración. Si incluye el parámetro opcional `ShowCacheNodeInfo`, la acción también devuelve los puntos de conexión de los nodos individuales del clúster.

Example

Para Memcached, el comando siguiente recupera el punto de conexión de configuración (`ConfigurationEndpoint`) y los puntos de conexión de nodos individuales (`Endpoint`) para el clúster de Memcached mycluster.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=DescribeCacheClusters
    &CacheClusterId=mycluster
    &ShowCacheNodeInfo=true
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
    &Version=2015-02-02
    &X-Amz-Credential=<credential>
```

> **Important**
>
> Si decide crear un CNAME para su punto de conexión de Memcached, a fin de que el cliente de detección automática reconozca el CNAME como un punto de conexión de configuración, deberá incluir `.cfg.` en el CNAME. Por ejemplo, incluya `mycluster.cfg.local` en el archivo php.ini en el parámetro `session.save_path`.

Búsqueda de puntos de conexión de grupos de replicación de Valkey o Redis OSS (API de ElastiCache)

Puede utilizar la API de ElastiCache para detectar los puntos de enlace de un grupo de reproducción y sus clústeres con la acción `DescribeReplicationGroups`. La acción devuelve el punto de conexión principal del grupo de reproducción y una lista de todos los clústeres del grupo de reproducción con sus puntos de enlace, junto con el punto de conexión de lector.

La operación siguiente recupera el punto de conexión principal (PrimaryEndpoint), el punto de conexión de lector (ReaderEndpoint) y los punto de conexión de nodo (ReadEndpoint) del grupo de reproducción `myreplgroup`. Use el punto de conexión principal para todas las operaciones de escritura.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=DescribeReplicationGroups
    &ReplicationGroupId=myreplgroup
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
    &Version=2015-02-02
    &X-Amz-Credential=<credential>
```

Para obtener más información, consulte DescribeReplicationGroups.

Uso de particiones en ElastiCache

Una partición (API/CLI: grupo de nodos) es una colección de entre uno y seis nodos de ElastiCache para Valkey o Redis OSS. Un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) nunca tendrá más de una partición. Con las particiones, puede separar las bases de datos de gran tamaño en unas partes más pequeñas, rápidas y fáciles de administrar denominadas particiones de datos. Esto puede aumentar la eficiencia de la base de datos al distribuir las operaciones en varias

secciones independientes. El uso de particiones puede tener muchas ventajas, como la mejora del rendimiento, la escalabilidad y la rentabilidad.

Puede crear un clúster con un mayor número de particiones y un menor número de réplicas con un total de hasta 90 nodos por clúster. Esta configuración de clúster puede variar desde 90 particiones y 0 réplicas hasta 15 particiones y 5 réplicas, que es el número máximo de réplicas permitido. Los datos del clúster están particionados en las distintas particiones del clúster. Si hay más de un nodo en una partición, esta implementa la reproducción con un nodo, siendo el nodo principal de lectura/escritura y los demás, nodos de réplica de solo lectura.

El límite de nodos o particiones se puede aumentar a un máximo de 500 por clúster si la versión del motor es Valkey 7.2 y posteriores o Redis OSS 5.0.6 a 7.1. Por ejemplo, puede elegir configurar un clúster de 500 nodos que oscila entre 83 particiones (uno primario y 5 réplicas por partición) y 500 particiones (único primario y sin réplicas). Asegúrese de que hay suficientes direcciones IP disponibles para acomodar el aumento. Algunos problemas comunes incluyen que las subredes del grupo de subredes tienen un rango CIDR demasiado pequeño o que otros clústeres comparten y utilizan considerablemente las subredes. Para obtener más información, consulte Creación de un grupo de subredes.

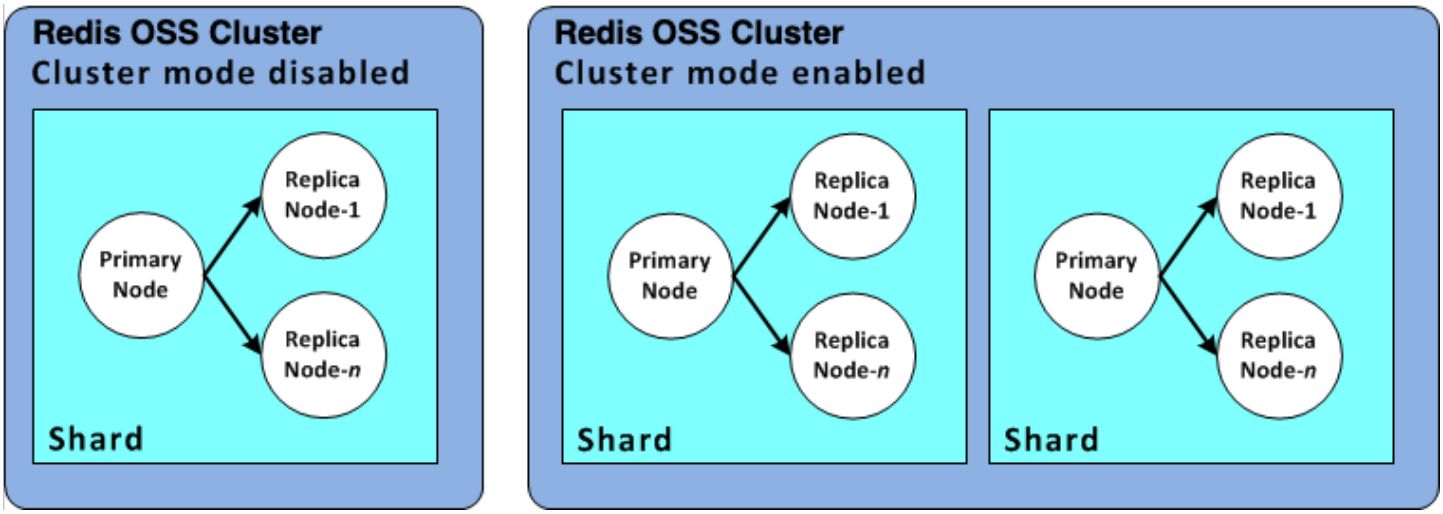
Para las versiones inferiores a 5.0.6, el límite es de 250 por clúster.

Para solicitar un aumento del límite, consulte AWS Service Limits y elija el tipo de límite Nodes per cluster per instance type (Nodos por clúster por tipo de instancias).

Cuando crea un clúster de Valkey o Redis OSS (modo de clúster habilitado) con la consola de ElastiCache, debe especificar el número de particiones del clúster y el número de nodos de las particiones. Para obtener más información, consulte Creación de un clúster de Valkey o Redis OSS (modo de clúster habilitado) (consola). Si utiliza la AWS CLI o la API de ElastiCache para crear un clúster (denominado grupo de reproducción en la API/CLI), puede configurar el número de nodos de una partición (API/CLI: grupo de nodos) de manera independiente. Para obtener más información, consulte los siguientes temas:

- API: CreateReplicationGroup
- CLI: create-replication-group

Los nodos de las particiones tienen las mismas especificaciones de memoria, almacenamiento y computación. La API de ElastiCache permite controlar los atributos de toda la partición, como el número de nodos, la configuración de seguridad y los periodos de mantenimiento del sistema.



Configuraciones de particiones de Valkey o Redis OSS

Para obtener más información, consulte Cambio de particiones sin conexión para Valkey o Redis OSS (modo de clúster habilitado) y Cambio de particiones en línea para Valkey o Redis OSS (modo de clúster habilitado).

Búsqueda del ID de una partición

Puede encontrar el ID de una partición con la Consola de administración de AWS, la AWS CLI o la API de ElastiCache.

Uso de Consola de administración de AWS

Temas

- Para Valkey o Redis OSS (modo de clúster deshabilitado)
- Para Valkey o Redis OSS (modo de clúster habilitado)

Para Valkey o Redis OSS (modo de clúster deshabilitado)

Los ID de partición del grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) son siempre `0001`.

Para Valkey o Redis OSS (modo de clúster habilitado)

En el siguiente procedimiento, se utiliza la Consola de administración de AWS para buscar un ID de partición del grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado).

Búsqueda del ID de partición en un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Valkey o Redis OSS y, a continuación, el nombre del grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado) cuyo ID de partición desea buscar.
3. En la columna de Shard Name (Nombre de la partición), el ID de partición lo forman los cuatro últimos dígitos del nombre de la partición.

Uso de AWS CLI

Para buscar los ID de partición (grupo de nodos) de los grupos de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) o Valkey o Redis OSS (modo de clúster habilitado), utilice la operación `describe-replication-groups` de la AWS CLI con el siguiente parámetro opcional.

- **`--replication-group-id`**—un parámetro opcional que, cuando se utiliza, limita los resultados de los detalles del grupo de reproducción especificado. Si se omite este parámetro, se devuelven los detalles de hasta 100 grupos de reproducción.

Example

Este comando devuelve los detalles de `sample-repl-group`.

Para Linux, macOS o Unix:

```
aws elasticache describe-replication-groups \
    --replication-group-id sample-repl-group
```

Para Windows:

```
aws elasticache describe-replication-groups ^
    --replication-group-id sample-repl-group
```

La salida de este comando es similar a la siguiente. Aquí están *resaltados* los ID de fragmento (grupo de nodos) para que resulte más fácil encontrarlos.

```
{
```

```
    "ReplicationGroups": [
        {
            "Status": "available",
            "Description": "2 shards, 2 nodes (1 + 1 replica)",
            "NodeGroups": [
                {
                    "Status": "available",
                    "Slots": "0-8191",
                    "NodeGroupId": "0001",
                    "NodeGroupMembers": [
                        {
                            "PreferredAvailabilityZone": "us-west-2c",
                            "CacheNodeId": "0001",
                            "CacheClusterId": "sample-repl-group-0001-001"
                        },
                        {
                            "PreferredAvailabilityZone": "us-west-2a",
                            "CacheNodeId": "0001",
                            "CacheClusterId": "sample-repl-group-0001-002"
                        }
                    ]
                },
                {
                    "Status": "available",
                    "Slots": "8192-16383",
                    "NodeGroupId": "0002",
                    "NodeGroupMembers": [
                        {
                            "PreferredAvailabilityZone": "us-west-2b",
                            "CacheNodeId": "0001",
                            "CacheClusterId": "sample-repl-group-0002-001"
                        },
                        {
                            "PreferredAvailabilityZone": "us-west-2a",
                            "CacheNodeId": "0001",
                            "CacheClusterId": "sample-repl-group-0002-002"
                        }
                    ]
                }
            ],
            "ConfigurationEndpoint": {
                "Port": 6379,
                "Address": "sample-repl-
group.9dcv5r.clustercfg.usw2.cache.amazonaws.com"
```

```
            },
            "ClusterEnabled": true,
            "ReplicationGroupId": "sample-repl-group",
            "SnapshotRetentionLimit": 1,
            "AutomaticFailover": "enabled",
            "SnapshotWindow": "13:00-14:00",
            "MemberClusters": [
                "sample-repl-group-0001-001",
                "sample-repl-group-0001-002",
                "sample-repl-group-0002-001",
                "sample-repl-group-0002-002"
            ],
            "CacheNodeType": "cache.m3.medium",
            "DataTiering": "disabled",
            "PendingModifiedValues": {}
        }
    ]
}
```

Uso de la API de ElastiCache

Para buscar los ID de partición (grupo de nodos) de los grupos de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) o Valkey o Redis OSS (modo de clúster habilitado), utilice la operación `describe-replication-groups` de la AWS CLI con el siguiente parámetro opcional.

- **`ReplicationGroupId`**—un parámetro opcional que, cuando se utiliza, limita los resultados de los detalles del grupo de reproducción especificado. Si se omite este parámetro, se devuelven los detalles de hasta *`xxx`* grupos de reproducción.

Example

Este comando devuelve los detalles de `sample-repl-group`.

Para Linux, macOS o Unix:

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DescribeReplicationGroup
   &ReplicationGroupId=sample-repl-group
   &Version=2015-02-02
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
```

```
&X-Amz-Credential=<credential>
```

Comparación de los clústeres Valkey, Memcached y Redis OSS basados en nodos

Amazon ElastiCache admite los motores de caché de Valkey, Memcached y Redis OSS. Cada motor proporciona algunas ventajas. Use la información de este tema para elegir el motor y la versión que mejor se adapte a sus necesidades.

> **Important**
>
> Tras crear un clúster de caché basado en nodos o un grupo de replicación, podrá actualizar el motor a una versión más reciente; sin embargo, no podrá volver a una versión inferior del motor. Si desea usar una versión más antigua del motor, deberá eliminar el clúster de caché basado en nodos o el grupo de replicación existente y crearlo de nuevo con la versión del motor más antigua.

En líneas generales, los motores son similares. Cada uno de ellos funciona como almacén de claves-valores en memoria. Sin embargo, en la práctica, existen diferencias importantes.

Elija Memcached si se dan las condiciones siguientes:

- Necesita el modelo más sencillo que sea posible.
- Necesita ejecutar nodos de gran tamaño con varios núcleos o subprocesos.
- Necesita tener la capacidad de escalar de forma ascendente y descendente, y agregar y eliminar nodos en función de la demanda del sistema.
- Necesita almacenar en la caché objetos.

Elija Valkey o Redis OSS con ElastiCache si se dan las siguientes condiciones:

- ElastiCache versión 7.2 para Valkey o versión 7.0 (mejorada) para Redis

  Desea utilizar funciones, publicación/suscripción en particiones o mejoras de ACL. Para obtener más información, consulte Redis OSS versión 7.0 (mejorada).
- ElastiCache versión 6.2 (mejorada) para Redis OSS

Desea tener la posibilidad de establecer niveles de datos entre memoria y SSD utilizando el tipo de nodo r6gd. Para obtener más información, consulte Organización de datos en niveles.

- ElastiCache versión 6.0 (mejorada) para Redis OSS

  Desea autenticar usuarios con el control de acceso basado en roles.

  Para obtener más información, consulte Redis OSS versión 6.0 (mejorada).

- ElastiCache versión 5.0.0 (mejorada) para Redis OSS

  Desea utilizar los flujos de Redis OSS, que son una estructura de datos de registro que permite a los productores añadir elementos nuevos en tiempo real y también permite a los consumidores consumir mensajes con o sin bloqueo.

  Para obtener más información, consulte Redis OSS versión 5.0.0 (mejorada).

- ElastiCache versión 4.0.10 (mejorada) para Redis OSS

  Admite tanto el cifrado como la adición o eliminación dinámica de particiones del clúster de Valkey o Redis OSS (modo de clúster habilitado).

  Para obtener más información, consulte Redis OSS versión 4.0.10 (mejorada).

Las siguientes versiones están en desuso, han llegado o están a punto de llegar al final de su vida útil.

- ElastiCache versión 3.2.10 (mejorada) para Redis OSS

  Admite la posibilidad de añadir o eliminar particiones de forma dinámica del clúster de Valkey o Redis OSS (modo de clúster habilitado).

  > **Important**
  >
  > En la actualidad, ElastiCache 3.2.10 para Redis no admite el cifrado.

  Para obtener más información, consulte los siguientes temas:

  - Versión 3.2.10 de Redis OSS (mejorada)

  - Prácticas recomendadas de cambios de particiones online para Redis OSS. Para obtener más información, consulte lo siguiente:
    - Prácticas recomendadas: cambios de los fragmentos online
    - Online Resharding and Shard Rebalancing for Valkey or Redis OSS (Cluster Mode Enabled)
  - Para obtener más información acerca del escalado de los clústeres de Redis OSS, consulte Escalado.

- ElastiCache versión 3.2.6 (mejorada) para Redis OSS

  Si necesita la funcionalidad de las versiones anteriores de Redis OSS, además de las características siguientes, elija 3.2.6:

  - Cifrado en tránsito. Para obtener más información, consulte Cifrado en tránsito de Amazon ElastiCache para Redis OSS.
  - Cifrado en reposo. Para obtener más información, consulte Cifrado en reposo de Amazon ElastiCache para Redis OSS.
- Versión 3.2.4 de ElastiCache para Redis OSS (modo de clúster habilitado)

  Si necesita la funcionalidad de 2.8.x además de las características siguientes, elija 3.2.4 (modo agrupado en clúster):

  - Es necesario que los datos se repartan entre dos a 500 grupos de nodos (solo en modo clúster).
  - Necesita indexación geoespacial (modo agrupado en clúster y no agrupado en clúster).
  - No necesita la compatibilidad con varias bases de datos.
- ElastiCache para Redis (modo no agrupado en clúster) 2.8.x y 3.2.4 (mejorado)

  Si se dan las condiciones siguientes, elija 2.8.x o 3.2.4 (en modo no agrupado en clúster):

  - Necesita tipos de datos complejos como, por ejemplo, cadenas, hash, listas, conjuntos, conjuntos clasificados mapas de bits.
  - Necesita ordenar o clasificar conjuntos de datos en memoria.
  - Necesita persistencia del almacén de claves.
  - Necesita replicar sus datos desde el nodo principal a una o varias réplicas de lectura para aplicaciones de lectura intensivas.
  - Necesita conmutación por error automática en caso de error en el nodo principal.
  - Necesita publicar y suscribirse a capacidades para informar a los clientes acerca de eventos en el servidor.

- Necesita capacidades de copia de seguridad y restauración para los clústeres basados en nodos, así como para las cachés sin servidor.
- Necesita compatibilidad con varias bases de datos.

Resumen comparativo de Memcached, Valkey o Redis OSS (modo de clúster deshabilitado) y Valkey o Redis OSS (modo de clúster habilitado)

	Memcached	Valkey o Redis OSS (modo de clúster deshabilitado)	Valkey o Redis OSS (modo de clúster habilitado)
Engine versions+	1.4.5 and later	4.0.10 and later	4.0.10 and later
Data types	Simple	2.8.x - Complex * Complex	3.2.x and later - Complex
Data partitioning	Yes	No	Yes
Cluster is modifiable	Yes	Yes	3.2.10 and later - Limited
Online resharding	No	No	3.2.10 and later
Encryption	in-transit 1.6.12 and later	4.0.10 and later	4.0.10 and later
Data tiering	No	6.2 and later	6.2 and later
Certificaciones de conformidad			
Compliance Certification			
FedRAMP	Sí - 1.6.12 y posteriores	4.0.10 y versiones posteriores	4.0.10 y versiones posteriores
HIPAA	Sí - 1.6.12 y posteriores	4.0.10 y versiones posteriores	4.0.10 y versiones posteriores
PCI DSS	Sí	4.0.10 y versiones posteriores	4.0.10 y versiones posteriores
Multi-threaded	Yes	No	No
Node type upgrade	No	Yes	Yes

	Memcached	Valkey o Redis OSS (modo de clúster deshabilitado)	Valkey o Redis OSS (modo de clúster habilitado)
Engine upgrading	Yes	Yes	Yes
High availability (replication)	No	Yes	Yes
Automatic failover	No	Optional	Required
Pub/Sub capabilities	No	Yes	Yes
Sorted sets	No	Yes	Yes
Backup and restore	For serverless caches only, not applicable to node-based clusters	Yes	Yes
Geospatial indexing	No	4.0.10 and later	Yes

Notas:

string, objects (like databases)

* string, sets, sorted sets, lists, hashes, bitmaps, hyperloglog

string, sets, sorted sets, lists, hashes, bitmaps, hyperloglog, geospatial indexes

+ Excludes versions which are deprecated, have reached or soon to reach end of life.

Tras elegir el motor para su clúster, recomendamos que use la versión más reciente de dicho motor. Para obtener más información, consulte Tipos de nodos compatibles.

Migración en línea para Valkey o Redis OSS

Al utilizar la migración en línea, puede migrar datos desde Valkey o Redis OSS de código abierto autoalojados en Amazon EC2 a Amazon ElastiCache.

Se refiere a la migración desde una instancia alojada de forma automática en el servicio de ElastiCache. Para obtener más información sobre la actualización de Redis OSS a Valkey en ElastiCache, consulte Actualización de las versiones del motor, incluidas las actualizaciones de varios motores.

> **Note**
>
> La migración en línea no es compatible con cachés sin servidor de ElastiCache o clústeres que se ejecutan en el tipo de nodo r6gd.

Descripción general

Para migrar datos desde Valkey o Redis OSS de código abierto autoalojados que se ejecutan en Amazon EC2 a Amazon ElastiCache, se necesita una implementación de Amazon ElastiCache existente o nueva. La implementación debe tener una configuración que esté preparada para el proceso de migración. También debe ajustarse a la configuración que desee, incluidos atributos como el tipo de instancia, el número de particiones y el número de réplicas.

La migración en línea está diseñada para la migración de datos desde Valkey o Redis OSS de código abierto autoalojados en Amazon EC2 a ElastiCache, y no para mover datos entre clústeres de ElastiCache.

> **Important**
>
> Recomendamos que lea las siguientes secciones en su totalidad antes de iniciar el proceso de migración online.

La migración comienza cuando llama a la operación de la API `StartMigration` o al comando AWS CLI. Al migrar clústeres de Valkey o Redis OSS en el modo de clúster deshabilitado, el proceso de migración convierte el nodo principal del clúster de ElastiCache Valkey o Redis OSS en una réplica de su nodo principal de Valkey o Redis OSS de origen. Al migrar clústeres de Valkey o Redis OSS en el modo de clúster habilitado, el proceso de migración convierte el nodo principal de cada partición de ElastiCache en una réplica de la partición correspondiente de su clúster de origen; además, posee las mismas ranuras.

Después de llevar a cabo los cambios en la parte del cliente, llame a la operación de la API `CompleteMigration`. Esta operación de API promueve la implementación de ElastiCache a

la implementación de Valkey o Redis OSS principal con nodos principales y de réplica (según corresponda). Ahora puede redirigir la aplicación de los clientes para comenzar a escribir datos en ElastiCache. Durante el proceso de migración, puede verificar el estado de la replicación si ejecuta el comando valkey-cli INFO en los nodos de Valkey y en los nodos principales de ElastiCache.

Pasos para realizar la migración

Los siguientes temas describen el proceso de la migración de datos:

- Preparación del origen y destino para la migración
- Prueba de la migración de datos
- Inicio de la migración
- Verificación del progreso de migración de datos
- Finalización de la migración de datos

Preparación del origen y destino para la migración

Con estos pasos, puede prepararse para migrar sus datos de un origen de Valkey o Redis autoalojado en EC2 a ElastiCache, o desde un clúster de Redis OSS a un clúster de ElastiCache Valkey.

Se refiere a la migración desde una instancia alojada de forma automática en el servicio de ElastiCache. Para obtener más información sobre la actualización de Redis OSS a Valkey en ElastiCache, consulte Actualización de las versiones del motor, incluidas las actualizaciones de varios motores.

Debe asegurarse de que se cumplen los cuatro requisitos previos que se mencionan a continuación, antes de comenzar la migración desde la consola de ElastiCache, la API o la CLI de AWS.

Preparación de los nodos de Valkey o Redis OSS de origen y destino para la migración

1. Identifique la implementación de ElastiCache de destino y asegúrese de que puede migrar los datos a ella.

   Una implementación nueva o existente de ElastiCache debe cumplir con los siguientes requisitos para efectuar la migración:

   - Utiliza Valkey o Redis OSS versión 5.0.6 o posterior.

- No tiene habilitado el cifrado en tránsito.
- Tiene habilitado Multi-AZ.
- Cuenta con suficiente memoria disponible para ajustar los datos desde el clúster de Valkey o Redis. Para configurar los ajustes adecuados de memoria reservada, consulte Administración de la memoria reservada para Valkey y Redis OSS.
- Para el modo de clúster deshabilitado, puede migrar directamente desde Valkey o Redis OSS 2.8.21 y versiones posteriores a la versión 5.0.6 de Valkey o Redis y posteriores si está utilizando la CLI o la versión 5.0.6 de Redis y posteriores con la CLI o la consola. Para el modo de clúster habilitado, puede migrar directamente desde cualquier versión de Valkey o Redis OSS con el modo de clúster habilitado a la versión de Redis OSS 5.0.6 y posteriores si está utilizando la CLI o la versión 5.0.6 de Redis OSS y posteriores con la CLI o la consola.
- Número de particiones en origen y destino coincidentes.
- No forma parte de un almacén de datos global.
- Tiene desactivada la agrupación de datos por niveles.

2. Asegúrese de que las configuraciones de su Valkey o Redis OSS de código abierto y la implementación de ElastiCache sean compatibles.

   Como mínimo, tienen que ser compatibles las siguientes características de la implementación de ElastiCache de destino con su configuración de Valkey o Redis OSS para realizar la replicación:

   - Su clúster no debe tener AUTH habilitado.
   - La configuración `protected-mode` debe estar establecida en no.
   - Si tiene la configuración `bind` en la configuración de Valkey o Redis OSS, debe actualizarla para permitir las solicitudes de los nodos de ElastiCache.
   - El número de bases de datos lógicas debe ser el mismo en el nodo de ElastiCache y en su clúster de Valkey o Redis OSS. Este valor se establece mediante `databases` en la configuración de Valkey o Redis OSS.
   - No se debe asignar un nuevo nombre a los comandos de Valkey o Redis OSS que llevan a cabo la modificación de datos para permitir que la replicación de los datos se realice correctamente; por ejemplo, `sync`, `psync`, `info`, `config`, `command` y `cluster`.
   - Para replicar los datos desde el clúster de Valkey o Redis OSS a ElastiCache, asegúrese de que haya suficiente memoria y CPU para gestionar esta carga adicional. Esta carga procede del archivo RDB que ha creado el clúster de Valkey o Redis OSS y se ha transferido por la red hasta el nodo de ElastiCache.

- Todas las instancias de Valkey o Redis OSS en el clúster de origen deben ejecutarse en el mismo puerto.

3. Asegúrese de que las instancias puedan conectarse con ElastiCache; para ello, haga lo siguiente:

   - Asegúrese de que la dirección IP de cada instancia sea privada.
   - Asigne o cree la implementación de ElastiCache en la misma nube privada virtual (VPC) que Valkey o Redis OSS en la instancia (recomendado).
   - Si las VPC son distintas, configure la interconexión de VPC para permitir el acceso entre los nodos. Para obtener más información acerca de la interconexión con VPC, consulte Patrones de acceso para obtener acceso a una caché de ElastiCache en una Amazon VPC.
   - El grupo de seguridad asociado a sus instancias de Valkey o Redis OSS debe permitir el tráfico de entrada desde los nodos de ElastiCache.

4. Asegúrese de que la aplicación pueda dirigir el tráfico a los nodos de ElastiCache después de que finalice la migración de datos. Para obtener más información, consulte Patrones de acceso para obtener acceso a una caché de ElastiCache en una Amazon VPC.

Prueba de la migración de datos

Después de completar todos los requisitos previos, puede validar la configuración de migración con la Consola de administración de AWS, la API de ElastiCache o la AWS CLI. El siguiente ejemplo muestra el uso de la CLI.

Pruebe la migración con el comando `test-migration` con los siguientes parámetros:

- `--replication-group-id`: el ID del grupo de replicación al que se van a migrar los datos.
- `--customer-node-endpoint-list`: lista de puntos de conexión desde los que deben migrarse los datos. La lista debe tener solo un elemento.

A continuación se muestra un ejemplo del uso de la CLI.

```
aws elasticache test-migration --replication-group-id test-cluster --customer-node-endpoint-list "Address='10.0.0.241',Port=6379"
```

ElastiCache validará la configuración de migración sin ninguna migración de datos real.

Inicio de la migración

Después de completar todos los requisitos previos, puede iniciar la migración de datos con la Consola de administración de AWS, la API de ElastiCache o la AWS CLI. Para el modo clúster activado, si la migración de ranura difiere, se realizará un cambio de particiones antes de la migración en activo. El siguiente ejemplo muestra el uso de la CLI.

> **Note**
>
> Recomendamos utilizar la API `TestMigration` para validar la configuración de la migración. Pero esto es estrictamente opcional.

Inicie la migración llamando al comando `start-migration` con los siguientes parámetros:

- `--replication-group-id`: identificador del grupo de reproducción de ElastiCache de destino
- `--customer-node-endpoint-list`: una lista de puntos de conexión con DNS o direcciones IP y el puerto en el que se ejecuta su clúster de Valkey o Redis OSS de origen. La lista solo puede tener un elemento tanto para el modo clúster desactivado como para el modo clúster activado. Si tiene habilitada la replicación encadenada, el punto de conexión puede apuntar a una réplica en vez de a un nodo principal en el clúster de Valkey o Redis OSS.

A continuación se muestra un ejemplo del uso de la CLI.

```
aws elasticache start-migration --replication-group-id test-cluster --customer-node-
endpoint-list "Address='10.0.0.241',Port=6379"
```

Al ejecutar este comando, el nodo principal de ElastiCache (en cada partición) se configura para convertirse en una réplica de su instancia de Valkey o Redis OSS (en la partición correspondiente que posee las mismas ranuras en el clúster habilitado para redis). El estado del clúster de ElastiCache cambia a Migrando y los datos comienzan a migrarse desde su instancia de Valkey o Redis OSS al nodo principal de ElastiCache. En función del tamaño de los datos y la carga en la instancia de Valkey o Redis OSS, la migración puede tardar un poco en finalizar. Puede comprobar el progreso de la migración si ejecuta el comando valkey-cli INFO en su instancia de Valkey o Redis OSS y en el nodo principal de ElastiCache.

Después de realizar la replicación correctamente, todo lo escrito en sus instancias de Valkey o Redis OSS se propaga al clúster de ElastiCache. Puede utilizar los nodos de ElastiCache para las lecturas.

Sin embargo, no es posible escribir en el clúster de ElastiCache. Si un nodo primario de ElastiCache tiene otros nodos de réplicas conectados, dichos nodos de réplica siguen replicándose desde el nodo primario de ElastiCache. De esta forma, todos los datos de su clúster de Valkey o Redis OSS se replican en todos los nodos del clúster de ElastiCache.

Si un nodo principal de ElastiCache no se puede convertir en una réplica de su instancia de Valkey o Redis OSS, se vuelve a intentar varias veces antes de transformarse de nuevo en principal. El estado del clúster de ElastiCache cambia a available (disponible) y se envía un evento del grupo de reproducción sobre el error para iniciar la migración. Para solucionar problemas como un error, compruebe lo siguiente:

- Busque el evento del grupo de reproducción. Utilice cualquier información específica del evento para arreglar el error de migración.
- Si el evento no ofrece ninguna información específica, asegúrese de que ha seguido las directrices que se indican en Preparación del origen y destino para la migración.
- Asegúrese de que la configuración de enrutamiento de la VPC y de las subredes permita el tráfico entre los nodos de ElastiCache y sus instancias de Valkey o Redis OSS.
- Asegúrese de que el grupo de seguridad asociado a sus instancias de Valkey o Redis OSS permita el tráfico de entrada desde los nodos de ElastiCache.
- Compruebe los registros de Valkey o Redis OSS de las instancias para obtener más información sobre los errores específicos de la replicación.

Verificación del progreso de migración de datos

Después de iniciar la migración de datos, puede hacer lo siguiente para realizar un seguimiento del progreso:

- Verifique que `master_link_status` de Valkey o Redis OSS es up en el comando `INFO` de los nodos principales de ElastiCache. También puede encontrar esta información en la consola de ElastiCache. Seleccione el clúster y, en CloudWatch metrics (Métricas de CloudWatch), observe el Primary Link Health Status (Estado del vínculo principal). Una vez que el valor alcanza 1, los datos se encuentran sincronizados.
- Puede comprobar si la réplica de ElastiCache tiene el estado En línea mediante la ejecución del comando `INFO` en sus instancias de Valkey o Redis OSS. Esta acción también ofrece información sobre el retardo de reproducción.

- Verifique el búfer de salida bajo del cliente mediante el comando CLIENT LIST en sus instancias de Valkey o Redis OSS.

Después de completar la migración de datos, los datos se encuentran sincronizados con cualquier escritura nueva que llega a los nodos principales de su clúster de Valkey o Redis OSS.

Finalización de la migración de datos

Cuando se encuentre preparado para realizar la migración total al clúster de ElastiCache, utilice el comando `complete-migration` de la CLI con los siguientes parámetros:

- `--replication-group-id`: el identificador del grupo de reproducción.
- `--force`: un valor que obliga a la migración a detenerse sin asegurarse de que los datos se encuentren sincronizados.

A continuación se muestra un ejemplo.

```
aws elasticache complete-migration --replication-group-id test-cluster
```

Al ejecutar este comando, el nodo principal de ElastiCache (en cada partición) deja de replicarse desde su instancia de Valkey o Redis OSS y lo promociona a principal. Esta promoción se completa normalmente en cuestión de minutos. Para confirmar la promoción a principal, compruebe el evento `Complete Migration successful for test-cluster`. En este punto, puede dirigir la aplicación a las lecturas y escrituras de ElastiCache. El estado del clúster de ElastiCache debe cambiar de migrating (migrando) a available (disponible).

Si se produce un error en la promoción a principal, el nodo principal de ElastiCache seguirá replicando desde su instancia de Valkey o Redis OSS. El clúster de ElastiCache sigue estando en el estado migrating (migrando) y se envía un mensaje de evento del grupo de reproducción acerca del error. Para solucionar este error, consulte lo siguiente:

- Compruebe el evento del grupo de reproducción. Utilice información específica del evento para arreglar el error.
- Puede que reciba un mensaje del evento sobre los datos no sincronizados. Si es así, asegúrese de que el principal de ElastiCache puede replicarse desde su instancia de Valkey o Redis OSS y de que ambos están sincronizados. Si todavía quiere detener la migración, puede ejecutar el comando anterior con la opción `–force`.

- Es posible que aparezca un mensaje de evento si se reemplaza uno de los nodos de ElastiCache. Puede volver a intentar el paso de migración completo después de que la sustitución finalice.

Realización de la migración de datos en línea con la consola

Puede utilizar la Consola de administración de AWS para migrar los datos desde su clúster al clúster de Valkey o Redis OSS.

Para realizar la migración de datos online utilizando la consola

1. Inicie sesión en la consola y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. Cree un nuevo clúster de Valkey o Redis OSS o elija uno existente. Asegúrese de que el clúster cumple los siguientes requisitos:
   - La versión del motor debe ser Valkey 7.2 y versiones posteriores, o Redis OSS 5.0.6 o posteriores.
   - Su clúster no debe tener AUTH habilitado.
   - La configuración `protected-mode` debe estar establecida en no.
   - Si tiene la configuración `bind` en la configuración de Valkey o Redis OSS, debe actualizarla para permitir las solicitudes de los nodos de ElastiCache.
   - El número de bases de datos debe ser el mismo en el nodo de ElastiCache y en su clúster de Valkey o Redis OSS. Este valor se establece mediante `databases` en la configuración del motor.
   - No se debe asignar un nuevo nombre a los comandos de Valkey o Redis OSS que llevan a cabo la modificación de datos para permitir que la replicación de los datos se realice correctamente.
   - Para replicar los datos desde el clúster de Valkey o Redis OSS a ElastiCache, asegúrese de que haya suficiente memoria y CPU para gestionar esta carga adicional. Esta carga procede del archivo RDB que ha creado el clúster de Valkey o Redis OSS y se ha transferido por la red hasta el nodo de ElastiCache.
   - El clúster tiene el estado available (disponible).
3. Con el clúster seleccionado, elija Migrate Data from Endpoint (Migrar datos desde el punto de enlace) para Actions (Acciones).

4. En el cuadro de diálogo Migrar datos desde el punto de conexión, introduzca la dirección IP y el puerto donde está disponible el clúster de Valkey o Redis OSS.

   > **Important**
   >
   > La dirección IP debe ser exacta. Si introduce una dirección que no es correcta, la migración da error.

5. Elija Start Migration (Iniciar migración).

   Cuando el clúster comienza la migración, cambia al estado Modifiying (Modificando) y, a continuación, a Migrating (Migrando).

6. Monitorice el proceso de migración al elegir Events (Eventos) en el panel de navegación.

Puede detener la migración en cualquier punto durante el proceso. Para hacerlo, elija el clúster y Stop Data Migration (Detener migración de datos) para Actions (Acciones). Después, el clúster pasa al estado Available (Disponible).

Si la migración se realiza correctamente, el clúster pasa al estado Available (Disponible) y el registro del evento muestra lo siguiente:

`Migration operation succeeded for replication group` *`ElastiCacheClusterName`*`.`

Si la migración falla, el clúster pasa al estado Available (Disponible) y el registro del evento muestra lo siguiente:

`Migration operation failed for replication group` *`ElastiCacheClusterName`*`.`

Elección de regiones y zonas de disponibilidad para ElastiCache

Puede proporcionar más escalabilidad y fiabilidad a sus clústeres de ElastiCache designando las regiones y las zonas de disponibilidad mediante el punto de conexión correspondiente.

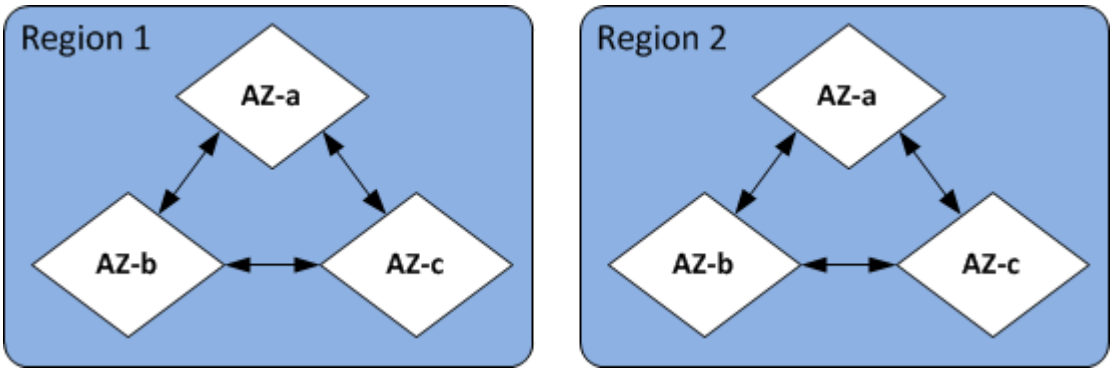
AWSLos recursos de informática en la nube de se alojan en centros de datos de alta disponibilidad. Para proporcionar escalabilidad y fiabilidad adicionales, estas instalaciones de centros de datos se encuentran en ubicaciones físicas diferentes. Dichas ubicaciones están categorizadas por regiones y zonas de disponibilidad.

AWSLas regiones de son de gran tamaño y se encuentran dispersas en distintas ubicaciones geográficas. Las zonas de disponibilidad son ubicaciones concretas dentro de una región de

AWS diseñadas para estar aisladas de los errores que se produzcan en las demás zonas de disponibilidad. Proporcionan conectividad de red económica y de baja latencia con las demás zonas de disponibilidad dentro de la misma región de AWS.

> **Important**
>
> Cada región es totalmente independiente. Cualquier actividad de ElastiCache que inicie (por ejemplo, la creación de clústeres) solo se ejecutará en la región predeterminada actual.

Para crear o trabajar con un clúster de una región específica, use el punto de conexión de servicio regional correspondiente. Para obtener información acerca de los puntos de conexión del servicio, consulte Regiones y puntos de conexión admitidos.



Regiones y zonas de disponibilidad de

Temas

- Consideraciones sobre las zonas de disponibilidad con Memcached
- Ubicación de los nodos
- Regiones y puntos de conexión admitidos
- Uso de Local Zones con ElastiCache
- Uso de Outposts con ElastiCache

Consideraciones sobre las zonas de disponibilidad con Memcached

La distribución de los nodos de Memcached en varias zonas de disponibilidad dentro de una región ayuda a proteger del impacto de un error catastrófico, como una pérdida de energía dentro de una zona de disponibilidad.

Almacenamiento en caché sin servidor

El almacenamiento en caché sin servidor de ElastiCache crea una caché de alta disponibilidad que abarca varias zonas de disponibilidad. Puede especificar subredes de distintas zonas de disponibilidad y la misma VPC al crear el clúster sin servidor; si no lo hace, ElastiCache elegirá automáticamente las subredes de la VPC predeterminada.

Diseño de su propio clúster de ElastiCache para Memcached

Un clúster de Memcached puede tener hasta 300 nodos. Al crear o agregar nodos al clúster de Memcached, puede especificar una única zona de disponibilidad para todos los nodos, permitir que ElastiCache elija una única zona de disponibilidad para todos los nodos, especificar las zonas de disponibilidad para cada nodo o permitir que ElastiCache elija una zona de disponibilidad para cada nodo. Los nodos nuevos se pueden crear en diferentes zonas de disponibilidad a medida que los agrega a un clúster de Memcached existente. Una vez que se crea un nodo de caché, la zona de disponibilidad no se puede modificar.

Si desea que un clúster de un único clúster de zona de disponibilidad tenga sus nodos distribuidos en varias zonas de disponibilidad, ElastiCache puede crear nodos nuevos en las distintas zonas de disponibilidad. A continuación, puede eliminar algunos o todos los nodos de caché originales. Recomendamos este enfoque.

Para migrar nodos de Memcached de una única zona de disponibilidad a varias zonas de disponibilidad

1. Modifique el clúster mediante nodos de caché nuevos en las zonas de disponibilidad donde los desea ubicar. En la solicitud, realice lo siguiente:

   - Establezca `AZMode` (CLI: `- -az-mode`) en `cross-az`.
   - Establezca `NumCacheNodes` (CLI: `- -num-cache-nodes`) al número de nodos de caché activos actualmente más el número de nodos de caché nuevos que desea crear.
   - Establezca `NewAvailabilityZones` (CLI: `- -new-availability-zones`) a una lista de las zonas en las que desea que se creen los nodos de caché nuevos. Para permitir que ElastiCache determine la zona de disponibilidad para cada nodo nuevo, no especifique una lista.
   - Establezca `ApplyImmediately` (CLI: `- -apply-immediately`) en true (verdadero).

> **Note**
>
> Si no utiliza la detección automática, asegúrese de actualizar la aplicación de cliente con los puntos de enlace de nodo de caché nuevos.

Antes de pasar al siguiente paso, asegúrese de que los nodos de Memcached se encuentren creados y disponibles.

2. Modifique el clúster al quitar los nodos que ya no desee en la zona de disponibilidad original. En la solicitud, realice lo siguiente:

   - Establezca `NumCacheNodes` (CLI: `- -num-cache-nodes`) al número de nodos de caché activos que desee después de aplicar esta modificación.
   - Establezca `CacheNodeIdsToRemove` (CLI: `- -nodes-to-remove`) a una lista de los nodos de caché que desea quitar del clúster.

     El número de los ID de los nodos de caché en la lista debe coincidir con el número de nodos activos en ese momento menos el valor en `NumCacheNodes`.
   - (Opcional) Establezca `ApplyImmediately` (CLI: `- -apply-immediately`) en true (verdadero).

     Si no establece `ApplyImmediately` (CLI: `- -apply-immediately`) en true (verdadero), las eliminaciones de nodos se llevarán a cabo en su próximo periodo de mantenimiento.

Ubicación de los nodos

Amazon ElastiCache permite ubicar todos los nodos de un clúster en una o varias zonas de disponibilidad (AZ). Además, si opta por ubicar los nodos en varias zonas de disponibilidad (opción recomendada), ElastiCache permite elegir la zona de disponibilidad de cada nodo o ElastiCache puede elegirlas en su nombre.

Al ubicar los nodos en distintas zonas de disponibilidad, se elimina la posibilidad de que un fallo, como un corte del suministro eléctrico, en una zona de disponibilidad provoque errores en todo el sistema. Las pruebas han demostrado que no hay ninguna diferencia de latencia importante entre ubicar todos los nodos en una zona de disponibilidad o repartirlos en varias zonas de disponibilidad.

Puede especificar una zona de disponibilidad para cada nodo cuando cree un clúster o agregando nodos cuando modifique un clúster existente. Al especificar una zona de disponibilidad para cada nodo al crear un clúster, la zona de disponibilidad debe estar disponible en ese grupo de subredes. Para obtener más información, consulte los siguientes temas:

- Creación de un clúster para Memcached
- Creación de un clúster para Valkey o Redis OSS
- Modificación de un clúster de ElastiCache
- Adición de nodos a un clúster de ElastiCache

Regiones y puntos de conexión admitidos

Amazon ElastiCache se encuentra disponible en varias regiones de AWS. Esto significa que puede lanzar clústeres de ElastiCache en las ubicaciones que cumplan sus requisitos. Por ejemplo, puede lanzarlos en la región de AWS más cercana a los clientes o en una región de AWS concreta que permita cumplir determinados requisitos legales.

Cada región se ha diseñado para que esté totalmente aislada de las demás regiones. Dentro de cada región hay varias zonas de disponibilidad (AZ). Las cachés de ElastiCache sin servidor replican automáticamente los datos en varias zonas de disponibilidad (excepto en `us-west-1`, donde los datos se replican en dos zonas de disponibilidad) para lograr un alto nivel de disponibilidad. Al diseñar su propio clúster de ElastiCache, puede optar por lanzar los nodos en diferentes zonas de disponibilidad a fin de obtener tolerancia a errores. Para obtener más información acerca de las regiones y zonas de disponibilidad, consulte Elección de regiones y zonas de disponibilidad para ElastiCache al comienzo de este tema.

Regiones en las que se admite ElastiCache

Nombre de la región/ Región	Punto de conexión	Protocolo	
Región del este de EE. UU. (Ohio) `us-east-2`	`elasticache.us-east-2.amazonaws.com`	HTTPS	
Región Este de EE. UU. (Norte de Virginia) `us-east-1`	`elasticache.us-east-1.amazonaws.com`	HTTPS	
Región Oeste de EE. UU. (Norte de California) `us-west-1`	`elasticache.us-west-1.amazonaws.com`	HTTPS	
Región del Oeste de EE. UU (Oregón) `us-west-2`	`elasticache.us-west-2.amazonaws.com`	HTTPS	
Región de Canadá (centro) `ca-central-1`	`elasticache.ca-central-1.amazonaws.com`	HTTPS	
Región Canadá (Oeste) `ca-west-1`	`elasticache.ca-west-1.amazonaws.com`	HTTPS	
Asia-Pacífico (Yakarta) `ap-southeast-3`	`elasticache.ap-southeast-3.amazonaws.com`	HTTPS	

Nombre de la región/ Región	Punto de conexión	Protocolo	
Región de Asia-Pacífico (Bombay) ap-south-1	elasticache.ap-south-1.amazonaws.com	HTTPS	
Región de Asia Pacífico (Hyderabad) ap-south-2	elasticache.ap-south-2.amazonaws.com	HTTPS	
Asia Pacífico (Tokio) ap-northeast-1	elasticache.ap-northeast-1.amazonaws.com	HTTPS	
Región de Asia-Pacífico (Seúl) ap-northeast-2	elasticache.ap-northeast-2.amazonaws.com	HTTPS	
Región Asia-Pacífico (Osaka) ap-northeast-3	elasticache.ap-northeast-3.amazonaws.com	HTTPS	
Región de Asia-Pacífico (Singapur) ap-southeast-1	elasticache.ap-southeast-1.amazonaws.com	HTTPS	
Región de Asia-Pacífico (Sídney) ap-southeast-2	elasticache.ap-southeast-2.amazonaws.com	HTTPS	

Nombre de la región/ Región	Punto de conexión	Protocolo	
Región de Europa (Fráncfort) `eu-central-1`	`elasticache.eu-central-1.amazonaws.com`	HTTPS	
Región Europa (Zúrich) `eu-central-2`	`elasticache.eu-central-2.amazonaws.com`	HTTPS	
Región Europa (Estocolmo) `eu-north-1`	`elasticache.eu-north-1.amazonaws.com`	HTTPS	
Región Medio Oriente (Baréin) `me-south-1`	`elasticache.me-south-1.amazonaws.com`	HTTPS	
Región de Oriente Medio (EAU) `me-central-1`	`elasticache.me-central-1.amazonaws.com`	HTTPS	
Región de Europa (Irlanda) `eu-west-1`	`elasticache.eu-west-1.amazonaws.com`	HTTPS	
Región de Europa (Londres) `eu-west-2`	`elasticache.eu-west-2.amazonaws.com`	HTTPS	

Nombre de la región/ Región	Punto de conexión	Protocolo	
Región EU (París) `eu-west-3`	`elasticache.eu-west-3.amazonaws.com`	HTTPS	
Región Europa (Milán) `eu-south-1`	`elasticache.eu-south-1.amazonaws.com`	HTTPS	
Región Europa (España) `eu-south-2`	`elasticache.eu-south-2.amazonaws.com`	HTTPS	
Región de América del Sur (São Paulo) `sa-east-1`	`elasticache.sa-east-1.amazonaws.com`	HTTPS	
Región China (Pekín) `cn-north-1`	`elasticache.cn-north-1.amazonaws.com.cn`	HTTPS	
Región China (Ningxia) `cn-northwest-1`	`elasticache.cn-northwest-1.amazonaws.com.cn`	HTTPS	
Región de Asia-Pacífico (Hong Kong) `ap-east-1`	`elasticache.ap-east-1.amazonaws.com`	HTTPS	
Región África (Ciudad del Cabo) `af-south-1`	`elasticache.af-south-1.amazonaws.com`	HTTPS	

Nombre de la región/ Región	Punto de conexión	Protocolo	
Región de Israel (Tel Aviv) `il-central-1`	`elasticache.il-central-1.amazonaws.com`	HTTPS	
AWS GovCloud (Oeste de EE. UU.) `us-gov-west-1`	`elasticache.us-gov-west-1.amazonaws.com`	HTTPS	
AWS GovCloud (Este de EE. UU.) `us-gov-east-1`	`elasticache.us-gov-east-1.amazonaws.com`	HTTPS	
Para obtener más información sobre el uso de AWS GovCloud (EE. UU.) con ElastiCache, consulte Servicios en la región de AWS GovCloud (EE. UU.).			

Algunas regiones admiten un subconjunto de tipos de nodo. Para ver una tabla de los tipos de nodos admitidos en cada región de AWS, consulte Tipos de nodos admitidos por región de AWS.

La mayoría de regiones permite establecer una conexión privada entre los puntos de conexión de la VPC y la API de ElastiCache mediante la creación de un punto de conexión de VPC de interfaz desde AWS PrivateLink. Para obtener más información, consulte Puntos de conexión de VPC de interfaz y API de ElastiCache (AWS PrivateLink).

Para obtener una tabla de los productos y servicios de AWS por región, consulte Productos y servicios por región.

Uso de Local Zones con ElastiCache

Una Zona local es una extensión de una región de AWS que está geográficamente cerca de sus usuarios. Puede ampliar cualquier nube privada virtual (VPC) de la región de AWS principal a Local Zones al crear una subred nueva y asignarla a la zona local. Cuando crea una subred en una zona

local, la VPC también se amplía a dicha zona local. La subred de la zona local funciona igual que otras subredes de su VPC.

Mediante el uso de Local Zones, puede colocar recursos como un clúster de ElastiCache en diferentes ubicaciones cerca de los usuarios.

Al crear un clúster de ElastiCache, puede elegir una subred en una zona local. Las zonas locales tienen sus propias conexiones a internet y admiten Direct Connect. Por lo tanto, los recursos creados en una zona local pueden prestar servicio a los usuarios locales con comunicaciones de muy baja latencia. Para obtener más información, consulte AWS Local Zones.

Una zona local se representa mediante un código de región de AWS seguido de un identificador que indica la ubicación, por ejemplo, `us-west-2-lax-1a`.

En este momento, las Local Zones disponibles son `us-west-2-lax-1a` y `us-west-2-lax-1b`.

Las siguientes limitaciones son aplicables a ElastiCache para Local Zones:

- No se admiten almacenes de datos globales.
- No se admite la migración en línea.
- Las Local Zones admiten los siguientes tipos de nodos:
  - Generación actual:

    Tipos de nodos M: `cache.m5.large`, `cache.m5.xlarge`, `cache.m5.2xlarge`, `cache.m5.4xlarge`, `cache.m5.12xlarge`, `cache.m5.24xlarge`

    Tipos de nodos R: `cache.r5.large`, `cache.r5.xlarge`, `cache.r5.2xlarge`, `cache.r5.4xlarge`, `cache.r5.12xlarge`, `cache.r5.24xlarge`

    Tipos de nodos T: `cache.t3.micro`, `cache.t3.small`, `cache.t3.medium`

Habilitación de una zona local

1. Habilite la zona local en la consola de Amazon EC2.

   Para obtener más información, consulte Habilitación de Local Zones en la Guía del usuario de Amazon EC2.

2. Cree una subred en la zona local.

Para obtener más información, consulte Creación de una subred en la VPC en la Guía del usuario de Amazon VPC.

3. Cree un grupo de subredes de ElastiCache en la zona local.

   Al crear un grupo de subredes de ElastiCache, elija el grupo de zonas de disponibilidad para la zona local.

   Para obtener más información, consulte Creación de un grupo de subredes.

4. Cree un clúster de ElastiCache for Memcached que utilice la subred de ElastiCache en la zona local.

   Para obtener más información, consulte Creación de un clúster de Memcached (consola).

5. Cree un clúster de ElastiCache para Redis OSS que utilice la subred de ElastiCache en la zona local. Para obtener más información, consulte uno de los siguientes temas:

   - Creación de un clúster de Valkey (modo de clúster deshabilitado) (consola)
   - Creación de un clúster de Valkey o Redis OSS (modo de clúster habilitado) (consola)

Uso de Outposts con ElastiCache

Puede usar AWS Outposts con ElastiCache. Outposts es un servicio completamente administrado que extiende la infraestructura, los servicios, las API y las herramientas de AWS a las instalaciones del cliente. Al proporcionar acceso local a la infraestructura administrada de AWS, AWS Outposts habilita a los clientes a crear y ejecutar aplicaciones en las instalaciones mediante el uso de las mismas interfaces de programación que en las regiones de AWS, a la vez que utilizan recursos informáticos y de almacenamiento locales para reducir la latencia y las necesidades de procesamiento de datos locales. Un Outpost es un grupo de capacidad informática y de almacenamiento de AWS implementada en un sitio del cliente. AWS opera, supervisa y administra esta capacidad como parte de una región de AWS. Puede crear subredes en Outpost y especificarlas al crear recursos de AWS como los clústeres de ElastiCache.

> **Note**
>
> En esta versión, se aplican las siguientes limitaciones:
>
> - ElastiCache para Outposts solo admite familias de nodos M5 y R5.
> - Multi-AZ (no se admite la reproducción cruzada de Outpost).

- La migración en directo no es compatible.
- No se admiten instantáneas locales.
- Los registros del motor y los registros lentos no se pueden habilitar.
- ElastiCache en Outposts no admite CoIP.
- ElastiCache para Outposts no se admite en las siguientes regiones: cn-north-1, cn-northwest-1 y ap-northeast-3.

Uso de Outposts con la consola de ElastiCache

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Cachés de Valkey, Cachés de Redis OSS o Cachés de Memcached.
3. Si ha elegido Cachés de Valkey, seleccione Crear caché de Valkey. Si ha elegido Cachés de Redis OSS, seleccione Crear caché de Redis OSS. Si ha elegido Cachés de Memcached, seleccione Crear caché de Memcached.
4. En Configuración del clúster, seleccione Diseñe su propia caché y Caché del clúster. Deje Modo de clúster en Desactivado. A continuación, cree un nombre y una descripción opcional para la caché.
5. En la ubicación, elija En las instalaciones.
6. En la sección En las instalaciones, verá el campo ID de Outpost. Introduzca el ID en el que se ejecutará el clúster.

   En todas las demás configuraciones de la Configuración del clúster, se pueden usar las opciones predeterminadas.
7. En Conectividad, seleccione Crear un nuevo grupo de subredes e introduzca el ID de la VPC. Deje la configuración predeterminada en el resto y elija Siguiente.

Configurar opciones en las instalaciones

Puede seleccionar un Outpost disponible para añadir el clúster o, si no hay Outposts disponibles, crear uno nuevo mediante los siguientes pasos:

En On-Premises options (Opciones en las instalaciones):

1. En Configuración de Valkey, Configuración de Redis OSS o Configuración de Memcached, según el motor que prefiera:

   a. Nombre: introduzca un nombre para el clúster.

   b. Descripción: introduzca una descripción para el clúster.

   c. Compatibilidad de versiones del motor: la versión del motor se basa en la región de AWS Outpost

   d. Puerto: para Valkey o Redis OSS, acepte el puerto 6379 predeterminado. Para Memcached, acepte el puerto predeterminado 11211. Si prefiere utilizar un puerto diferente, escriba el número de puerto.

   e. Grupo de parámetros: utilice el menú desplegable para seleccionar un grupo de parámetros predeterminado o personalizado.

   f. Tipo de nodo: las instancias disponibles se basan en la disponibilidad de Outposts. Si utiliza Valkey o Redis OSS, el Asistente de portabilidad para .NET para Outposts solo admite las familias de nodos M5 y R5. En la lista desplegable, seleccione Outposts y, a continuación, seleccione el tipo de nodo disponible que desee utilizar para este clúster. A continuación, seleccione Guardar.

   g. Número de réplicas: ingrese el número de réplicas de lectura que desea crear para este grupo de reproducción. Debe tener al menos una y no más de cinco réplicas de lectura. El valor predeterminado es 2.

      Los nombres generados de forma automática de las réplicas de lectura siguen el mismo patrón que el nombre del clúster principal, con un guion y un número secuencial de tres dígitos agregados al final, que comienza por `-002`. Por ejemplo, si el grupo de reproducción se denomina `MyGroup`, entonces los nombres de los secundarios serían `MyGroup-002`, `MyGroup-003`, `MyGroup-004`, `MyGroup-005`, `MyGroup-006`.

2. En Conectividad:

   a. Grupo de subredes: en la lista, seleccione Create new (Crear nuevo).

      - Name: ingrese un nombre para el grupo de la subred.
      - Descripción: ingrese una descripción para el grupo de la subred.
      - ID de VPC: el ID de VPC debe coincidir con la VPC de Outpost. Si selecciona una VPC que no tiene ID de subred en los Outposts, la lista se devolverá vacía.

- Zona de disponibilidad o Outpost: seleccione el Outpost que utiliza.
- ID de subred: seleccione un ID de subred que se encuentre disponible para el Outpost. Si no hay ID de subred disponibles, debe crearlos. Para obtener más información, consulte Creación de una subred.

b. Seleccione Crear.

Visualización de detalles del clúster de Outpost

En la página de la lista, seleccione un clúster que pertenezca a un Outpost de AWS y tenga en cuenta lo siguiente al ver los Detalles del clúster:

- Zona de disponibilidad: esto representará el Outpost, con un ARN (nombre de recurso de Amazon) y el número de recurso de AWS.
- Nombre del Outpost: el nombre del Outpost de AWS.

Uso de Outposts con la CLI de AWS

Puede usar la AWS Command Line Interface (AWS CLI) para controlar varios servicios de AWS desde la línea de comandos y automatizarlos mediante scripts. Puede utilizar la AWS CLI para operaciones ad hoc (únicas).

Descarga y configuración de la AWS CLI

La AWS CLI se ejecuta en Windows, macOS o Linux. Utilice el siguiente procedimiento para descargarlo y configurarlo.

Para descargar, instalar y configurar la CLI

1. Descargue la CLI de AWS en la página web de Interfaz de línea de comandos de AWS.
2. Siga las instrucciones para Instalar la AWS CLI y Configurar la AWS CLI en la Guía del usuario de AWS Command Line Interface

Uso de la CLI de AWS con Outposts

Utilice la siguiente operación de la CLI para crear un clúster que utilice Outposts:

- create-cache-cluster: con esta operación, el parámetro `outpost-mode` acepta un valor que especifica si los nodos del clúster se crean en un único Outpost o en varios Outposts.

> **Note**
>
> En este momento, solo el modo `single-outpost` es compatible.

```
aws elasticache create-cache-cluster \
   --cache-cluster-id cluster id \
   --outpost-mode single-outpost \
```

Uso de ElastiCache

En esta sección, encontrará detalles sobre cómo administrar los diferentes componentes de la implementación de ElastiCache.

Temas

- Instantánea y restauración
- Versiones del motor y actualización en ElastiCache
- Procedimientos recomendados y estrategias de almacenamiento en caché de ElastiCache
- Administración del clúster basado en nodos en ElastiCache
- Conexión automática de una instancia de EC2 y una caché de ElastiCache
- Escalado de ElastiCache
- Introducción a los filtros de Bloom
- Introducción a la búsqueda vectorial
- Introducción a JSON para Valkey y Redis OSS
- Etiquetado de los recursos de ElastiCache
- Uso del enfoque Well-Architected de Amazon ElastiCache
- Prácticas recomendadas y pasos comunes de solución de problemas con ElastiCache

Instantánea y restauración

Las cachés de Amazon ElastiCache que ejecutan Valkey, Redis OSS o Memcached sin servidor pueden realizar copias de seguridad de sus datos mediante la creación de una instantánea. Puede utilizar la copia de seguridad para restaurar una caché o para propagar los datos iniciales de una nueva caché. La copia de seguridad se compone de los metadatos de la caché, junto con todos los datos de la caché. Todas las copias de seguridad se escriben en Amazon Simple Storage Service (Amazon S3), lo que proporciona un almacenamiento duradero. Puede restaurar sus datos en cualquier momento creando una nueva caché de Valkey, Redis OSS o Memcached sin servidor y rellenándola con los datos de una copia de seguridad. Con ElastiCache, puede administrar copias de seguridad mediante la Consola de administración de AWS, la AWS Command Line Interface (AWS CLI) y la API de ElastiCache.

Si está pensando en eliminar la caché y es importante conservar los datos, puede tomar una medida de precaución adicional. Para ello, cree primero una copia de seguridad manual, compruebe que

su estado sea disponible y elimine la caché. Con este procedimiento, se asegura de que seguirá disponiendo de los datos de la caché aunque haya un error en la copia de seguridad. Puede volver a intentar realizar una copia de seguridad siguiendo las prácticas recomendadas descritas anteriormente.

Temas

- Limitaciones del proceso de copia de seguridad
- Impacto en el rendimiento de las copias de seguridad de los clústeres basados en nodos
- Programación de copias de seguridad automáticas
- Copias de seguridad manuales
- Creación de una copia de seguridad final
- Descripción de copias de seguridad
- Copiar copias de seguridad
- Exportación de una copia de seguridad
- Restauración desde una copia de seguridad a una nueva caché
- Eliminación de una copia de seguridad
- Etiquetado de copias de seguridad
- Tutorial: propagación de un nuevo clúster basado en nodos con una copia de seguridad creada externamente

Limitaciones del proceso de copia de seguridad

Debe tener en cuenta las limitaciones siguientes a la hora de planear o realizar copias de seguridad:

- La copia de seguridad y restauración solo es compatible con las cachés que se ejecutan en Valkey, Redis OSS o Memcached sin servidor.
- Para los clústeres de Valkey o Redis OSS (modo de clúster deshabilitado), la copia de seguridad y la restauración no son compatibles con los nodos `cache.t1.micro`. Todos los demás tipos de nodos de caché son compatibles con la característica.
- Para los clústeres de Valkey o Redis OSS (modo de clúster habilitado), la copia de seguridad y la restauración son compatibles con todos los tipos de nodos.
- Durante 24 horas continuas, no podrá crear más de 24 copias de seguridad manuales por caché sin servidor. En el caso de los clústeres basados en nodos de Valkey y Redis OSS, no podrá crear más de 20 copias de seguridad manuales por cada nodo del clúster.

- Valkey o Redis OSS (modo de clúster habilitado) solo permiten realizar copias de seguridad en el clúster (para la API o la CLI, en el grupo de replicación). Valkey o Redis OSS (modo de clúster habilitado) no permiten realizar copias de seguridad en las particiones (para la API o la CLI, en el grupo de nodos).
- Durante el proceso de copia de seguridad, no podrá realizar operaciones de la API o la CLI en la caché sin servidor. Puede realizar operaciones de la API o la CLI en un clúster basado en nodos durante la copia de seguridad.
- Si utiliza cachés de Valkey o Redis OSS con organización de datos en niveles, no puede exportar una copia de seguridad a Amazon S3.
- Puede restaurar una copia de seguridad de un clúster utilizando el tipo de nodo r6gd solo en clústeres que utilicen el tipo de nodo r6gd.

Impacto en el rendimiento de las copias de seguridad de los clústeres basados en nodos

Las copias de seguridad en las cachés sin servidor son transparentes para la aplicación y no afectan al rendimiento. Sin embargo, al crear copias de seguridad para clústeres basados en nodos, el rendimiento puede verse afectado en cierta medida en función de la memoria reservada disponible. Las copias de seguridad de los clústeres basados en nodos no están disponibles con ElastiCache para Memcached, pero sí con ElastiCache para Redis OSS.

A continuación, se ofrecen algunas pautas para mejorar el rendimiento de la realización de copias de seguridad para clústeres basados en nodos.

- Defina el parámetro `reserved-memory-percent`: para mitigar el exceso de paginación, recomendamos que configure el parámetro reserved-memory-percent. Este parámetro impide que Valkey o Redis OSS consuma toda la memoria disponible del nodo y puede ayudar a reducir la cantidad de páginas creadas. También puede ver mejoras de rendimiento simplemente usando un nodo de mayor tamaño. Para obtener más información sobre los parámetros reserved-memory y reserved-memory-percent, consulte Administración de la memoria reservada para Valkey y Redis OSS.

- Cree copias de seguridad a partir de una réplica de lectura: si ejecuta Valkey o Redis OSS en un grupo de nodos con más de un nodo, puede tomar una copia de seguridad del nodo principal o una de las réplicas de lectura. Dados los orígenes del sistema requeridos durante BGSAVE,

recomendamos que cree copias de seguridad de una de las réplicas de lectura. Aunque la copia de seguridad se está creando a partir de la réplica, el nodo principal no se ve afectado por los requisitos de recursos del comando BGSAVE. El nodo principal puede continuar atendiendo a las solicitudes sin ralentizarse.

Para hacer esto, consulte Creación de una copia de seguridad manual (consola) y en el campo Cluster Name (Nombre del clúster) en la ventana Create Backup (Crear copia de seguridad), elija una réplica en lugar del nodo primario predeterminado.

Si elimina un grupo de reproducción y solicita una copia de seguridad final, ElastiCache siempre realizará la copia de seguridad del nodo primario. De este modo, se garantiza que se capturan los datos de Valkey o Redis OSS más recientes antes de eliminar el grupo de replicación.

Programación de copias de seguridad automáticas

Puede habilitar copias de seguridad automáticas para cualquier caché sin servidor de Valkey o Redis OSS o para cualquier clúster basado en nodos. Cuando se habilitan las copias de seguridad automáticas, ElastiCache crea una copia de seguridad de la caché una vez al día. Esto no afecta a la caché y el cambio es inmediato. Las copias de seguridad automáticas pueden ayudarle a protegerse frente a la pérdida de datos. En caso de error, puede crear una nueva caché y restaurar los datos de la copia de seguridad más reciente. El resultado es una caché iniciada en activo; es decir, ya está cargada con sus datos y lista para el uso. Para obtener más información, consulte Restauración desde una copia de seguridad a una nueva caché.

Puede habilitar copias de seguridad automáticas para cualquier caché sin servidor de Memcached. Cuando se habilitan las copias de seguridad automáticas, ElastiCache crea una copia de seguridad de la caché una vez al día. Esto no afecta a la caché y el cambio es inmediato. Las copias de seguridad automáticas pueden ayudarle a protegerse frente a la pérdida de datos. En caso de error, puede crear una nueva caché y restaurar los datos de la copia de seguridad más reciente. El resultado es una caché iniciada en activo; es decir, ya está cargada con sus datos y lista para el uso. Para obtener más información, consulte Restauración desde una copia de seguridad a una nueva caché.

Al programar copias de seguridad automáticas, debe planificar los ajustes siguientes:

- Hora de inicio de la copia de seguridad: hora del día en la que ElastiCache comienza a crear una copia de seguridad. Puede configurar la ventana de copia de seguridad en el momento que le resulte más conveniente. Si no especifica un periodo de copia de seguridad, ElastiCache asignará uno de forma automática.

- Límite de retención de copia de seguridad: número de días que se retiene la copia de seguridad en Amazon S3. Por ejemplo, si establece el límite de retención en 5, una copia de seguridad que se realice hoy se conservaría durante 5 días. Al finalizar el límite de retención, la copia de seguridad se eliminará automáticamente.

  El límite máximo de retención de copia de seguridad es de 35 días. Si el límite de retención de copia de seguridad se establece en 0, las copias de seguridad se deshabilitarán en la caché.

Al programar copias de seguridad automáticas, ElastiCache empezará a crear la copia de seguridad. Puede configurar la ventana de copia de seguridad en el momento que le resulte más conveniente. Si no especifica un periodo de copia de seguridad, ElastiCache asignará uno de forma automática.

Puede habilitar o deshabilitar las copias de seguridad automáticas al crear una nueva caché o al actualizar una caché existente mediante la consola de ElastiCache, la AWS CLI o la API de ElastiCache. En el caso de Valkey y Redis OSS, esto se hace marcando la casilla Habilitar copias de seguridad automáticas en las secciones Configuración avanzada de Valkey o Configuración avanzada de Redis OSS. En el caso de Memcached, esto se hace marcando la casilla Habilitar copias de seguridad automáticas en la sección Configuración avanzada de Memcached.

Copias de seguridad manuales

Además de las copias de seguridad automáticas, puede crear una copia de seguridad manual en cualquier momento. A diferencia de las copias de seguridad automáticas, que se eliminan automáticamente después de un periodo de retención determinado, las copias de seguridad manuales no tienen periodo de retención que determine su eliminación automática. Aunque elimine la memoria caché, se conservarán todas las copias de seguridad manuales de esa memoria caché. Si ya no desea conservar una copia de seguridad manual, deberá eliminarla de forma explícita.

Además de crear una copia de seguridad manual, puede crear copias de seguridad manuales de las maneras siguientes:

- Copiar copias de seguridad. No importa si la copia de seguridad de origen se creó automáticamente o manualmente.
- Creación de una copia de seguridad final. Cree una copia de seguridad inmediatamente antes de eliminar un clúster o nodo.

Puede crear una copia de seguridad manual de una caché mediante la Consola de administración de AWS, la AWS CLI o la API de ElastiCache.

Puede generar copias de seguridad manuales a partir de réplicas que estén habilitadas para el modo de clúster y deshabilitadas para el modo de clúster.

Creación de una copia de seguridad manual (consola)

Para crear una copia de seguridad de una caché (consola)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Cachés de Valkey, Cachés de Redis OSS o Cachés de Memcached, según sean sus preferencias.
3. Elija la casilla situada a la izquierda del nombre de la caché de la que desea realizar la copia de seguridad.
4. Elija Backup (Copia de seguridad).
5. En el cuadro de diálogo Create Backup, escriba un nombre para la copia de seguridad en el cuadro Backup Name. Recomendamos que el nombre indique que el clúster proviene de una copia de seguridad y especifique la fecha y la hora en que se creó la copia de seguridad.

Las restricciones para la asignación de nombres de clúster son las siguientes:

- Deben contener entre 1 y 40 caracteres alfanuméricos o guiones.
- Deben comenzar por una letra.
- No pueden contener dos guiones consecutivos.
- No pueden terminar con un guion.

6. Elija Create Backup.

   El estado del clúster cambia a snapshotting.

Creación de una copia de seguridad manual (AWS CLI)

Copia de seguridad manual de una caché sin servidor con la AWS CLI

Para crear una copia de seguridad manual de una caché mediante la AWS CLI, use la operación `create-serverless-snapshot` de la AWS CLI con los parámetros siguientes:

- `--serverless-cache-name`: el nombre de la memoria caché sin servidor cuya copia de seguridad está realizando.
- `--serverless-cache-snapshot-name`: nombre de la instantánea que se creará.

Para Linux, macOS o Unix:

```
aws elasticache create-serverless-snapshot \
                        --serverless-cache-name CacheName \
                        --serverless-cache-snapshot-name bkup-20231127
```

Para Windows:

```
aws elasticache create-serverless-snapshot ^
    --serverless-cache-name CacheName ^
    --serverless-cache-snapshot-name bkup-20231127
```

Copia de seguridad manual de un clúster basado en nodos con la AWS CLI

Para crear una copia de seguridad manual de un clúster basado en nodos mediante la AWS CLI, use la operación `create-snapshot` de la AWS CLI con los parámetros siguientes:

- `--cache-cluster-id`
  - Si el clúster que está copiando no tiene nodos de réplica, `--cache-cluster-id` será el nombre del clúster cuya copia de seguridad está creando; por ejemplo, *`mycluster`*.
  - Si el clúster que está copiando tiene uno o varios nodos de réplica, `--cache-cluster-id` será el nombre del nodo del clúster que desea usar para la copia de seguridad. Por ejemplo, el nombre puede ser *`mycluster-002`*.

  Utilice este parámetro solo para realizar copias de seguridad de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado).

- `--replication-group-id`: nombre del clúster (CLI/API: grupo de replicación) de Valkey o Redis OSS (modo de clúster habilitado) que se utilizará como origen de la copia de seguridad. Utilice este parámetro para realizar copias de seguridad de un clúster de Valkey o Redis OSS (modo de clúster habilitado).

- `--snapshot-name`: nombre de la instantánea que se creará.

  Las restricciones para la asignación de nombres de clúster son las siguientes:
  - Deben contener entre 1 y 40 caracteres alfanuméricos o guiones.
  - Deben comenzar por una letra.
  - No pueden contener dos guiones consecutivos.
  - No pueden terminar con un guion.

Ejemplo 1: Creación de una copia de seguridad de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) que no tiene nodos de réplica

La siguiente operación de la AWS CLI crea la copia de seguridad `bkup-20150515` a partir del clúster `myNonClusteredRedis` de Valkey o Redis OSS (modo de clúster deshabilitado) que no tiene réplicas de lectura.

Para Linux, macOS o Unix:

```
aws elasticache create-snapshot \
    --cache-cluster-id myNonClusteredRedis \
    --snapshot-name bkup-20150515
```

Para Windows:

```
aws elasticache create-snapshot ^
    --cache-cluster-id myNonClusteredRedis ^
    --snapshot-name bkup-20150515
```

Ejemplo 2: Creación de una copia de seguridad de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) con nodos de réplica

La siguiente operación de la AWS CLI crea la copia de seguridad `bkup-20150515` a partir del clúster `myNonClusteredRedis` de Valkey o Redis OSS (modo de clúster deshabilitado). Esta copia de seguridad dispone de una o más réplicas de lectura.

Para Linux, macOS o Unix:

```
aws elasticache create-snapshot \
    --cache-cluster-id myNonClusteredRedis-001 \
    --snapshot-name bkup-20150515
```

Para Windows:

```
aws elasticache create-snapshot ^
    --cache-cluster-id myNonClusteredRedis-001 ^
    --snapshot-name bkup-20150515
```

Ejemplo de resultados: Copia de seguridad de un clúster de Valkey o Redis OSS (modo clúster deshabilitado) con nodos de réplica

La salida de la operación será similar a lo que se indica a continuación.

```
{
    "Snapshot": {
        "Engine": "redis",
        "CacheParameterGroupName": "default.redis6.x",
        "VpcId": "vpc-91280df6",
```

```
        "CacheClusterId": "myNonClusteredRedis-001",
        "SnapshotRetentionLimit": 0,
        "NumCacheNodes": 1,
        "SnapshotName": "bkup-20150515",
        "CacheClusterCreateTime": "2017-01-12T18:59:48.048Z",
        "AutoMinorVersionUpgrade": true,
        "PreferredAvailabilityZone": "us-east-1c",
        "SnapshotStatus": "creating",
        "SnapshotSource": "manual",
        "SnapshotWindow": "08:30-09:30",
        "EngineVersion": "6.0",
        "NodeSnapshots": [
            {
                "CacheSize": "",
                "CacheNodeId": "0001",
                "CacheNodeCreateTime": "2017-01-12T18:59:48.048Z"
            }
        ],
        "CacheSubnetGroupName": "default",
        "Port": 6379,
        "PreferredMaintenanceWindow": "wed:07:30-wed:08:30",
        "CacheNodeType": "cache.m3.2xlarge",
        "DataTiering": "disabled"
    }
}
```

Ejemplo 3: Creación de una copia de seguridad de un clúster de Valkey o Redis OSS (modo de clúster habilitado)

La siguiente operación de la AWS CLI crea la copia de seguridad `bkup-20150515` a partir del clúster `myClusteredRedis` de Valkey o Redis OSS (modo de clúster habilitado). Tenga en cuenta que debe usar el `--replication-group-id` en lugar del `--cache-cluster-id` para identificar el origen. Tenga en cuenta que ElastiCache realiza la copia de seguridad con el nodo de réplica, si existe, o con el nodo principal si no hay ningún nodo de réplica disponible.

Para Linux, macOS o Unix:

```
aws elasticache create-snapshot \
    --replication-group-id myClusteredRedis \
    --snapshot-name bkup-20150515
```

Para Windows:

```
aws elasticache create-snapshot ^
    --replication-group-id myClusteredRedis ^
    --snapshot-name bkup-20150515
```

Ejemplo de resultados: Creación de una copia de seguridad de un clúster de Valkey o Redis OSS (modo de clúster habilitado)

La salida de esta operación será similar a lo que se indica a continuación.

```
{
    "Snapshot": {
        "Engine": "redis",
        "CacheParameterGroupName": "default.redis6.x.cluster.on",
        "VpcId": "vpc-91280df6",
        "NodeSnapshots": [
            {
                "CacheSize": "",
                "NodeGroupId": "0001"
            },
            {
                "CacheSize": "",
                "NodeGroupId": "0002"
            }
        ],
        "NumNodeGroups": 2,
        "SnapshotName": "bkup-20150515",
        "ReplicationGroupId": "myClusteredRedis",
        "AutoMinorVersionUpgrade": true,
        "SnapshotRetentionLimit": 1,
        "AutomaticFailover": "enabled",
        "SnapshotStatus": "creating",
        "SnapshotSource": "manual",
        "SnapshotWindow": "10:00-11:00",
        "EngineVersion": "6.0",
        "CacheSubnetGroupName": "default",
        "ReplicationGroupDescription": "2 shards 2 nodes each",
        "Port": 6379,
        "PreferredMaintenanceWindow": "sat:03:30-sat:04:30",
        "CacheNodeType": "cache.r3.large",
        "DataTiering": "disabled"
    }
}
```

Temas relacionados

Para obtener más información, consulte create-snapshot en la Referencia de comandos de la AWS CLI.

Creación de una copia de seguridad con CloudFormation

Puede utilizar CloudFormation para crear una copia de seguridad de la caché de ElastiCache Redis OSS o Valkey con las propiedades `AWS::ElastiCache::ServerlessCache` o `AWS::ElastiCache::ReplicationGroup`.

Uso del recurso **`AWS::ElastiCache::ServerlessCache`**

Utilice esta opción para crear una copia de seguridad con el recurso AWS::ElastiCache::ServerlessCache:

```
Resources:
                    iotCatalog:
                        Type: AWS::ElastiCache::ServerlessCache
                            Properties:
                            ...
                            ServerlessCacheName: "your-cache-name"
                            Engine: "redis"
                            CacheUsageLimits
```

Uso del recurso AWS::ElastiCache::ReplicationGroup

Utilice el recurso `AWS::ElastiCache::ReplicationGroup`:

```
Resources:
                    iotCatalog:
                        Type: AWS::ElastiCache::ReplicationGroup
                            Properties:
                            ...
                            ReplicationGroupDescription: "Description of your
 replication group"
                            Engine: "redis"
                            CacheNodeType
                            NumCacheClusters
                            AutomaticFailoverEnabled
                            AtRestEncryptionEnabled
```

Creación de una copia de seguridad final

Puede crear una copia de seguridad final con la consola de ElastiCache, la AWS CLI o la API de ElastiCache.

Creación de una copia de seguridad final (consola)

Puede crear una copia de seguridad final al eliminar una caché sin servidor de Valkey, Memcached o Redis OSS, o un clúster basado en nodos de Valkey o Redis OSS mediante la consola de ElastiCache.

Para crear una copia de seguridad final al eliminar una caché, seleccione Sí en el cuadro de diálogo de eliminación, bajo Crear copia de seguridad, y póngale un nombre a la copia de seguridad.

Temas relacionados

- Uso de Consola de administración de AWS
- Eliminación de un grupo de reproducción (consola)

Creación de una copia de seguridad final (AWS CLI)

Puede crear una copia de seguridad final al eliminar una caché de Redis usando la AWS CLI.

Temas

- Al eliminar una caché de Valkey, una caché sin servidor de Memcached o una caché de Redis OSS
- Al eliminar un clúster de Valkey o Redis OSS sin réplicas de lectura
- Al eliminar un clúster de Valkey o Redis OSS con réplicas de lectura

Al eliminar una caché de Valkey, una caché sin servidor de Memcached o una caché de Redis OSS

Para crear una copia de seguridad final, use la operación de la `delete-serverless-cache` AWS CLI con los parámetros siguientes.

- `--serverless-cache-name`: nombre de la caché que va a eliminar.
- `--final-snapshot-name`: nombre de la copia de seguridad.

El siguiente código crea la copia de seguridad final `bkup-20231127-final` al eliminar la caché `myserverlesscache`.

Para Linux, macOS o Unix:

```
aws elasticache delete-serverless-cache \
        --serverless-cache-name myserverlesscache \
        --final-snapshot-name bkup-20231127-final
```

Para Windows:

```
aws elasticache delete-serverless-cache ^
        --serverless-cache-name myserverlesscache ^
        --final-snapshot-name bkup-20231127-final
```

Para obtener más información, consulte delete-serverless-cache en la Referencia de comandos de la AWS CLI.

Al eliminar un clúster de Valkey o Redis OSS sin réplicas de lectura

Para crear una copia de seguridad final para un clúster basado en nodos sin réplicas de lectura, utilice la operación `delete-cache-cluster` de la AWS CLI con los siguientes parámetros.

- `--cache-cluster-id`: nombre del clúster que va a eliminar.
- `--final-snapshot-identifier`: nombre de la copia de seguridad.

El siguiente código crea la copia de seguridad final `bkup-20150515-final` al eliminar el clúster `myRedisCluster`.

Para Linux, macOS o Unix:

```
aws elasticache delete-cache-cluster \
        --cache-cluster-id myRedisCluster \
        --final-snapshot-identifier bkup-20150515-final
```

Para Windows:

```
aws elasticache delete-cache-cluster ^
        --cache-cluster-id myRedisCluster ^
```

```
        --final-snapshot-identifier bkup-20150515-final
```

Para obtener más información, consulte delete-cache-cluster en la Referencia de comandos de la AWS CLI.

Al eliminar un clúster de Valkey o Redis OSS con réplicas de lectura

Para crear una copia de seguridad final al eliminar un grupo de reproducción, use la operación de la `delete-replication-group` AWS CLI con los parámetros siguientes:

- `--replication-group-id`: nombre del grupo de replicación que se va a eliminar.
- `--final-snapshot-identifier`: nombre de la copia de seguridad final.

El siguiente código realiza la copia de seguridad final `bkup-20150515-final` al eliminar el grupo de reproducción `myReplGroup`.

Para Linux, macOS o Unix:

```
aws elasticache delete-replication-group \
        --replication-group-id myReplGroup \
        --final-snapshot-identifier bkup-20150515-final
```

Para Windows:

```
aws elasticache delete-replication-group ^
        --replication-group-id myReplGroup ^
        --final-snapshot-identifier bkup-20150515-final
```

Para obtener más información, consulte describe-replication-group en la Referencia de comandos de la AWS CLI.

Descripción de copias de seguridad

Los procedimientos siguientes muestran cómo mostrar una lista de sus copias de seguridad. Si lo desea, también puede ver los detalles de una copia de seguridad determinada.

Descripción de copias de seguridad (consola)

Para mostrar copias de seguridad mediante la Consola de administración de AWS

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, seleccione Backups (Copias de seguridad).
3. Para ver los detalles de una copia de seguridad determinada, elija la casilla situada a la izquierda del nombre de la copia de seguridad.

Descripción de las copias de seguridad sin servidor (AWS CLI)

Para mostrar una lista de las copias de seguridad sin servidor y, de forma opcional, los detalles de una copia de seguridad específica, use la operación `describe-serverless-cache-snapshots` de la CLI.

Ejemplos

La operación siguiente usa el parámetro `--max-records` para mostrar hasta 20 copias de seguridad asociadas a su cuenta. La omisión del parámetro `--max-records` permite mostrar hasta 50 copias de seguridad.

```
aws elasticache describe-serverless-cache-snapshots --max-records 20
```

La operación siguiente usa el parámetro `--serverless-cache-name` para mostrar solo las copias de seguridad asociadas a la caché `my-cache`.

```
aws elasticache describe-serverless-cache-snapshots --serverless-cache-name my-cache
```

La operación siguiente usa el parámetro `--serverless-cache-snapshot-name` para mostrar los detalles de la copia de seguridad `my-backup`.

```
aws elasticache describe-serverless-cache-snapshots --serverless-cache-snapshot-
name my-backup
```

Para obtener más información, consulte describe-serverless-cache-snapshots en la Referencia de comandos de la AWS CLI.

Descripción de las copias de seguridad de clústeres basados en nodos (AWS CLI)

Para mostrar una lista de las copias de seguridad de clústeres basados en nodos y, de forma opcional, los detalles específicos de una copia de seguridad, use la operación de la CLI `describe-snapshots`.

Ejemplos

La operación siguiente usa el parámetro `--max-records` para mostrar hasta 20 copias de seguridad asociadas a su cuenta. La omisión del parámetro `--max-records` permite mostrar hasta 50 copias de seguridad.

```
aws elasticache describe-snapshots --max-records 20
```

La operación siguiente usa el parámetro `--cache-cluster-id` para mostrar solo las copias de seguridad asociadas al clúster `my-cluster`.

```
aws elasticache describe-snapshots --cache-cluster-id my-cluster
```

La operación siguiente usa el parámetro `--snapshot-name` para mostrar los detalles de la copia de seguridad `my-backup`.

```
aws elasticache describe-snapshots --snapshot-name my-backup
```

Para obtener más información, consulte describe-snapshots en la Referencia de comandos de la AWS CLI.

Copiar copias de seguridad

Puede realizar una copia de cualquier copia de seguridad, independientemente de si se creó de forma automática o manual. También puede exportar su copia de seguridad para que pueda acceder a ella desde fuera de ElastiCache. Para obtener instrucciones acerca de cómo exportar su copia de seguridad, consulte Exportación de una copia de seguridad.

Los procedimientos siguientes muestran cómo copiar una copia de seguridad.

Copia de copias de seguridad (consola)

Para copiar una copia de seguridad (consola)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. Para ver una lista de las copias de seguridad, en el panel de navegación izquierdo, elija Backups (Copias de seguridad).
3. En la lista de copias de seguridad, active la casilla situada a la izquierda del nombre de la copia de seguridad que desea copiar.
4. Seleccione Acciones y Copiar.
5. En el cuadro New backup name (Nombre del nuevo backup), escriba un nombre para la nueva copia de seguridad.
6. Elija Copiar.

Copia de una copia de seguridad sin servidor (AWS CLI)

Para copiar una copia de seguridad de una caché sin servidor, utilice la operación `copy-serverless-cache-snapshot`.

Parámetros

- `--source-serverless-cache-snapshot-name`: nombre de la copia de seguridad que se copiará.
- `--target-serverless-cache-snapshot-name`: nombre de la copia de seguridad.

El ejemplo siguiente realiza una copia de una copia de seguridad automática.

Para Linux, macOS o Unix:

```
aws elasticache copy-serverless-cache-snapshot \
    --source-serverless-cache-snapshot-name automatic.my-cache-2023-11-27-03-15 \
    --target-serverless-cache-snapshot-name my-backup-copy
```

Para Windows:

```
aws elasticache copy-serverless-cache-snapshot ^
    --source-serverless-cache-snapshot-name automatic.my-cache-2023-11-27-03-15 ^
    --target-serverless-cache-snapshot-name my-backup-copy
```

Para obtener más información, consulte `copy-serverless-cache-snapshot` en la AWS CLI.

Copia de una copia de seguridad del clúster basado en nodos (AWS CLI)

Para copiar una copia de seguridad de un clúster basado en nodos, utilice la operación `copy-snapshot`.

Parámetros

- `--source-snapshot-name`: nombre de la copia de seguridad que se copiará.
- `--target-snapshot-name`: nombre de la copia de seguridad.
- `--target-bucket`: reservado para la exportación de la copia de seguridad. No use este parámetro cuando realice una copia de una copia de seguridad. Para obtener más información, consulte Exportación de una copia de seguridad.

El ejemplo siguiente realiza una copia de una copia de seguridad automática.

Para Linux, macOS o Unix:

```
aws elasticache copy-snapshot  \
    --source-snapshot-name automatic.my-redis-primary-2014-03-27-03-15 \
    --target-snapshot-name amzn-s3-demo-bucket
```

Para Windows:

```
aws elasticache copy-snapshot ^
    --source-snapshot-name automatic.my-redis-primary-2014-03-27-03-15 ^
    --target-snapshot-name amzn-s3-demo-bucket
```

Para obtener más información, consulte `copy-snapshot` en la AWS CLI.

Exportación de una copia de seguridad

Amazon ElastiCache permite exportar su copia de seguridad de ElastiCache para Redis OSS a un bucket de Amazon Simple Storage Service (Amazon S3), lo que facilita el acceso a la copia de seguridad desde fuera de ElastiCache. Puede exportar una copia de seguridad con la consola de ElastiCache, la AWS CLI o la API de ElastiCache.

Exportar una copia de seguridad puede resultar útil si necesita lanzar un clúster en otra región de AWS. De este modo, puede exportar sus datos en una sola región de AWS, copiar el archivo .rdb a la nueva región de AWS y, a continuación, utilizar dicho archivo .rdb para inicializar los datos en la nueva caché, en lugar de esperar a que el nuevo clúster se rellene durante el uso. Para obtener información acerca de la propagación de datos en un nuevo clúster, consulte Tutorial: propagación de un nuevo clúster basado en nodos con una copia de seguridad creada externamente. Otro motivo por el que es posible que desee exportar los datos de su caché es para usar el archivo .rdb para el procesamiento sin conexión.

Important

- La copia de seguridad de ElastiCache y el bucket de Amazon S3 en el que desea copiar el contenido deben encontrarse en la misma región de AWS.

  Aunque las copias de seguridad copiadas en un bucket de Amazon S3 se encuentran cifradas, recomendamos encarecidamente que no conceda acceso a otras personas al bucket de Amazon S3 en el que desea almacenar las copias de seguridad.
- La exportación de una copia de seguridad a Amazon S3 no se admite en clústeres que utilizan la organización de datos en niveles. Para obtener más información, consulte Organización de datos en niveles en ElastiCache.
- La exportación de una copia de seguridad está disponible para: clústeres de Valkey basados en nodos, clústeres de Redis OSS basados en nodos y cachés sin servidor de Valkey, Memcached y Redis OSS. No se puede exportar una copia de seguridad para los clústeres de Memcached basados en nodos.

Para poder exportar una copia de seguridad a un bucket de Amazon S3, debe disponer de un bucket de Amazon S3 en la misma región de AWS que la copia de seguridad. Conceda acceso a ElastiCache al bucket. Los primeros dos pasos muestran cómo realizar esto último.

Crear un bucket de Amazon S3

Los siguientes pasos utilizan la consola de Amazon S3 para crear un bucket de Amazon S3 al que se exporta, y en el que se almacena, la copia de seguridad de ElastiCache.

Creación de un bucket de Amazon S3

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. Seleccione la opción Crear bucket.
3. En Create a Bucket - Select a Bucket Name and Region, haga lo siguiente:

    a. En Bucket Name (Nombre del bucket), escriba un nombre para el bucket de Amazon S3.

    El nombre del bucket de Amazon S3 debe estar conforme con DNS. De lo contrario, ElastiCache no podrá acceder al archivo de copia de seguridad. Las reglas para la conformidad con DNS son:

    - Los nombres deben tener un mínimo de 3 y un máximo de 63 caracteres de largo.
    - Los nombres deben ser una serie de una o más etiquetas separadas por un punto (.) en el que cada etiqueta:
        - Comienza por una letra minúscula o un número.
        - Termina con una letra minúscula o un número.
        - Solo contiene letras minúsculas, números y guiones.
    - Los nombres no pueden tener el formato de una dirección IP (por ejemplo, 192.0.2.0).

    b. En la lista de Region (Región), elija una región de AWS para su bucket de Amazon S3. Esta región de AWS debe encontrarse en la misma región de AWS que la de la copia de seguridad de ElastiCache que desea exportar.

    c. Seleccione Crear.

Para obtener más información sobre la creación de un bucket de Amazon S3, consulte la sección de Creación de un bucket en la Guía del usuario de Amazon Simple Storage Service.

Concesión de acceso a ElastiCache a su bucket de Amazon S3

Para que ElastiCache pueda copiar una instantánea en un bucket de Amazon S3, debe actualizar su política de bucket de IAM para conceder acceso a ElastiCache al bucket.

> **Warning**
>
> Aunque las copias de seguridad copiadas en un bucket de Amazon S3 se encuentran cifradas, cualquier persona que tenga acceso al bucket de Amazon S3 puede acceder a sus datos. Por tanto, recomendamos encarecidamente que establezca políticas de IAM para impedir el acceso no autorizado a este bucket de Amazon S3. A fin de obtener más información, consulte Administración del acceso en la Guía del usuario de Amazon S3.

Para crear los permisos adecuados en un bucket de Amazon S3, siga los pasos que se describen a continuación.

Para conceder acceso a ElastiCache al bucket S3.

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. Elija el nombre del bucket de Amazon S3 en el que desea copiar la copia de seguridad. Este debe ser el bucket de S3 que creó en Crear un bucket de Amazon S3.
3. Elija la pestaña Permissions (Permisos) y debajo Permissions (Permisos), elija Access control list (ACL) (Lista de control de acceso (ACL)) y luego Edit (Editar).
4. Agregue la identificación canónica `540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353` con las siguientes opciones:

   - Objetos: List (Lista) o Write (Escribir
   - ACL de bucket: Read (Lectura) o Write (Escritura

   > **Note**
   >
   > - Para la región PDT de GovCloud, el ID canónico es `40fa568277ad703bd160f66ae4f83fc9dfdfd06c2f1b5060ca22442ac3ef8be6`.
   > - Para la región OSU de GovCloud, el ID canónico es `c54286759d2a83da9c480405349819c993557275cf37d820d514b42da6893f5c`.

5. Seleccione Save.

Exportación de una copia de seguridad de ElastiCache

Ya ha creado el bucket de S3 y concedido permisos de ElastiCache para acceder a él. A continuación, podrá utilizar la consola de ElastiCache, la AWS CLI o la API de ElastiCache para exportar su instantánea a ella.

A continuación, se muestra un ejemplo del aspecto que tendría la política actualizada.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "Policy15397346",
    "Statement": [
        {
            "Sid": "Stmt15399484",
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::hkg-elasticache-backup",
                "arn:aws:s3:::hkg-elasticache-backup/*"
            ]
        }
    ]
}
```

A continuación, se muestra un ejemplo del aspecto que tendría la política de IAM actualizada para el bucket de S3 de las regiones a las que se suscribió. (En este ejemplo se utiliza la región Asia Pacífico (Hong Kong).

JSON

```
{
    "Version":"2012-10-17",
    "Id": "Policy15397346",
    "Statement": [
        {
            "Sid": "Stmt15399483",
```

```
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::hkg-elasticache-backup",
                "arn:aws:s3:::hkg-elasticache-backup/*"
            ]
        },
        {
            "Sid": "Stmt15399484",
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::hkg-elasticache-backup",
                "arn:aws:s3:::hkg-elasticache-backup/*"
            ]
        }
    ]
}
```

Exportación de una copia de seguridad de ElastiCache (consola)

Los siguientes pasos utilizan la consola de ElastiCache para exportar una copia de seguridad a un bucket de Amazon S3 a fin de que pueda tener acceso a la copia de seguridad desde fuera de ElastiCache. El bucket de Amazon S3 debe encontrarse en la misma región de AWS que la copia de seguridad de ElastiCache.

Para exportar una copia de seguridad de ElastiCache a un bucket de Amazon S3

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. Para ver una lista de las copias de seguridad, en el panel de navegación izquierdo, elija Backups (Copias de seguridad).
3. De la lista de copias de seguridad, active la casilla situada a la izquierda del nombre de la copia de seguridad que desea exportar.
4. Elija Copiar.
5. En Create a Copy of the Backup? (¿Desea crear una copia del backup?), haga lo siguiente:

    a. En el cuadro New backup name (Nombre del nuevo backup), escriba un nombre para la nueva copia de seguridad.

El nombre debe tener entre 1 y 1 000 caracteres y debe admitir la codificación UTF-8.

ElastiCache agrega un identificador de instancias y `.rdb` al valor que ingrese aquí. Por ejemplo, si ingresa `my-exported-backup`, ElastiCache creará `my-exported-backup-0001.rdb`.

b. Desde la lista de Target S3 Location (Ubicación de S3 de destino), elija el nombre del bucket de Amazon S3 al que desea copiar la copia de seguridad (el bucket que creó en Crear un bucket de Amazon S3).

La Target S3 Location (Ubicación de S3 de destino) debe ser un bucket de Amazon S3 de la misma región de AWS que la de la copia de seguridad con los siguientes permisos para que el proceso de exportación se realice correctamente.

- Acceso al objeto: Read (Lectura) y Write (Escritura).
- Permisos de acceso: lectura.

Para obtener más información, consulte Concesión de acceso a ElastiCache a su bucket de Amazon S3.

c. Elija Copiar.

> **Note**
>
> Si su bucket de S3 no tiene los permisos necesarios para que ElastiCache pueda exportar una copia de seguridad, recibirá uno de los siguientes mensajes de error. Vuelva a Concesión de acceso a ElastiCache a su bucket de Amazon S3 para añadir los permisos especificados y vuelva a intentar exportar la copia de seguridad.
>
> - ElastiCache has not been granted READ permissions %s on the S3 Bucket.
>
>   Solución: añada los permisos Read en el bucket.
> - ElastiCache has not been granted WRITE permissions %s on the S3 Bucket.
>
>   Solución: añada los permisos Write en el bucket.
> - ElastiCache has not been granted READ_ACP permissions %s on the S3 Bucket.
>
>   Solución: añada Read como permiso de acceso en el bucket.

Si desea copiar su copia de seguridad en otra región de AWS, utilice Amazon S3 para copiarla. Para obtener más información, consulte Copia de objetos en la Guía del usuario de Amazon Simple Storage Service.

Exportación de una copia de seguridad sin servidor de ElastiCache (AWS CLI)

Exportación de una copia de seguridad de una caché sin servidor

Exporte la copia de seguridad a un bucket de Amazon S3 con la operación de la CLI `export-serverless-cache-snapshot` con los siguientes parámetros:

Parámetros

- `--serverless-cache-snapshot-name`: nombre de la copia de seguridad que se copiará.
- `--s3-bucket-name`: escriba el nombre del bucket de Amazon S3 donde desea exportar la copia de seguridad. Se realizará una copia de la copia de seguridad en el bucket especificado.

  El `--s3-bucket-name` debe ser un bucket de Amazon S3 de la misma región de AWS que la de la copia de seguridad con los siguientes permisos para que el proceso de exportación se realice correctamente.
  - Acceso al objeto: Read (Lectura) y Write (Escritura).
  - Permisos de acceso: lectura.

La operación siguiente permite copiar una copia de seguridad en my-s3-bucket.

Para Linux, macOS o Unix:

```
aws elasticache export-serverless-cache-snapshot \
    --serverless-cache-snapshot-name automatic.my-redis-2023-11-27 \
    --s3-bucket-name my-s3-bucket
```

Para Windows:

```
aws elasticache export-serverless-cache-snapshot ^
    --serverless-cache-snapshot-name automatic.my-redis-2023-11-27 ^
    --s3-bucket-name my-s3-bucket
```

Exportación de una copia de seguridad de clústeres de ElastiCache basados en nodos (AWS CLI)

Exportación de una copia de seguridad de un clúster basado en nodos

Exporte la copia de seguridad a un bucket de Amazon S3 con la operación de la CLI `copy-snapshot` con los siguientes parámetros:

Parámetros

- `--source-snapshot-name`: nombre de la copia de seguridad que se copiará.
- `--target-snapshot-name`: nombre de la copia de seguridad.

  El nombre debe tener entre 1 y 1 000 caracteres y debe admitir la codificación UTF-8.

  ElastiCache agrega un identificador de instancias y `.rdb` al valor que ingrese aquí. Por ejemplo, si ingresa `my-exported-backup`, ElastiCache creará `my-exported-backup-0001.rdb`.
- `--target-bucket`: escriba el nombre del bucket de Amazon S3 donde desea exportar la copia de seguridad. Se realizará una copia de la copia de seguridad en el bucket especificado.

  El `--target-bucket` debe ser un bucket de Amazon S3 de la misma región de AWS que la de la copia de seguridad con los siguientes permisos para que el proceso de exportación se realice correctamente.
  - Acceso al objeto: Read (Lectura) y Write (Escritura).
  - Permisos de acceso: lectura.

  Para obtener más información, consulte Concesión de acceso a ElastiCache a su bucket de Amazon S3.

La operación siguiente permite copiar una copia de seguridad en my-s3-bucket.

Para Linux, macOS o Unix:

```
aws elasticache copy-snapshot \
    --source-snapshot-name automatic.my-redis-primary-2016-06-27-03-15 \
    --target-snapshot-name my-exported-backup \
    --target-bucket my-s3-bucket
```

Para Windows:

```
aws elasticache copy-snapshot ^
    --source-snapshot-name automatic.my-redis-primary-2016-06-27-03-15 ^
    --target-snapshot-name my-exported-backup ^
    --target-bucket my-s3-bucket
```

Restauración desde una copia de seguridad a una nueva caché

Puede restaurar una copia de seguridad existente desde Valkey en una nueva caché de Valkey o en un clúster basado en nodos, y restaurar una copia de seguridad de Redis OSS existente en una nueva caché de Redis OSS o en un clúster basado en nodos. También puede restaurar una copia de seguridad de una caché sin servidor de Memcached existente en una nueva caché sin servidor de Memcached.

Restauración de una copia de seguridad en una caché sin servidor (consola)

> **Note**
>
> ElastiCache sin servidor admite archivos RDB compatibles con Valkey 7.2 y versiones posteriores, y versiones de Redis OSS entre la 5.0 y la versión más reciente disponible.

Para restaurar una copia de seguridad en una caché sin servidor (consola)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, seleccione Backups (Copias de seguridad).
3. En la lista de copias de seguridad, active la casilla situada a la izquierda del nombre de la copia de seguridad que desea restaurar.
4. Elija Acciones y, a continuación, Restaurar.
5. Escriba un nombre para la nueva caché sin servidor y, si lo desea, una descripción.
6. Haga clic en Crear para crear la nueva caché e importar los datos de la copia de seguridad.

Restauración de una copia de seguridad en un clúster basado en nodos (consola)

Cómo restaurar una copia de seguridad en un clúster basado en nodos (consola)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, seleccione Backups (Copias de seguridad).
3. En la lista de copias de seguridad, active la casilla situada a la izquierda del nombre de la copia de seguridad que desea restaurar.
4. Elija Acciones y, a continuación, Restaurar.

5. Elija Caché basada en nodos y personalice la configuración del clúster con, por ejemplo, el tipo de nodo, los tamaños, la cantidad de particiones, las réplicas, la ubicación en las zonas de disponibilidad y la configuración de seguridad.
6. Elija Crear para crear el nuevo clúster basado en nodos e importar los datos de la copia de seguridad.

Restauración de una copia de seguridad en una caché sin servidor (AWS CLI)

> **Note**
>
> ElastiCache sin servidor admite archivos RDB compatibles con Valkey 7.2 y versiones posteriores, y versiones de Redis OSS entre la 5.0 y la versión más reciente disponible.

Para restaurar una copia de seguridad en una caché sin servidor (AWS CLI)

En el siguiente ejemplo de la AWS CLI, se crea una nueva caché mediante `create-serverless-cache` y se importan los datos de una copia de seguridad.

Para Linux, macOS o Unix:

```
aws elasticache create-serverless-cache \

    --serverless-cache-name CacheName \
    --engine redis
    --snapshot-arns-to-restore Snapshot-ARN
```

Para Windows:

```
aws elasticache create-serverless-cache ^

    --serverless-cache-name CacheName ^
    --engine redis ^
    --snapshot-arns-to-restore Snapshot-ARN
```

Eliminación de una copia de seguridad

Las copias de seguridad automáticas se eliminan automáticamente cuando finaliza el límite de retención. Si elimina un clúster, también se eliminarán todas sus copias de seguridad automáticas. Si

elimina un grupo de reproducción, también se eliminarán todas las copias de seguridad automáticas de los clústeres de dicho grupo.

ElastiCache ofrece una operación de la API de eliminación que permite eliminar copias de seguridad en cualquier momento, independientemente de si la copia de seguridad se creó de forma automática o manual. Dado que las copias de seguridad manuales no tienen límite de retención, estas copias solo se pueden eliminar de forma manual.

Puede eliminar una copia de seguridad con la consola de ElastiCache, la AWS CLI o la API de ElastiCache.

Eliminación de una copia de seguridad (consola)

El siguiente procedimiento permite eliminar una copia de seguridad mediante la consola de ElastiCache.

Para eliminar una copia de seguridad

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, seleccione Copias de seguridad.

   Aparecerá la pantalla Backups (Copias de seguridad) con una lista de sus copias de seguridad.
3. Elija la casilla situada a la izquierda del nombre de la copia de seguridad que desee eliminar.
4. Elija Eliminar.
5. Si desea eliminar este backup, seleccione Delete en la pantalla de confirmación Delete Backup. El estado cambia a deleting.

Eliminación de una copia de seguridad sin servidor (AWS CLI)

Use la operación delete-snapshot de la AWS CLI con el siguiente parámetro para eliminar una copia de seguridad sin servidor.

- `--serverless-cache-snapshot-name`: nombre de la copia de seguridad que se va a eliminar.

El código siguiente elimina la copia de seguridad `myBackup`.

```
aws elasticache delete-serverless-cache-snapshot --serverless-cache-snapshot-
name myBackup
```

Para obtener más información, consulte delete-serverless-cache-snapshot en la Referencia de comandos de la AWS CLI.

Eliminación de una copia de seguridad del clúster basado en nodos (AWS CLI)

Use la operación delete-snapshot de la AWS CLI con el parámetro siguiente para eliminar una copia de seguridad del clúster basado en nodos.

- `--snapshot-name`: nombre de la copia de seguridad que se va a eliminar.

El código siguiente elimina la copia de seguridad `myBackup`.

```
aws elasticache delete-snapshot --snapshot-name myBackup
```

Para obtener más información, consulte delete-snapshot en la Referencia de comandos de la AWS CLI.

Etiquetado de copias de seguridad

Puede asignar sus propios metadatos a cada copia de seguridad en forma de etiquetas. Las etiquetas permiten clasificar las copias de seguridad de diversas maneras, por ejemplo, según su finalidad, propietario o entorno. Esto es útil cuando tiene muchos recursos del mismo tipo: puede identificar rápidamente un recurso específico en función de las etiquetas que le haya asignado. Para obtener más información, consulte Recursos que se pueden etiquetar.

Las etiquetas de asignación de costos permiten realizar un seguimiento de los costos de varios servicios de AWS al agrupar los gastos de las facturas por valores de etiqueta. Para obtener más información sobre las etiquetas de asignación de costos, consulte Uso de etiquetas de asignación de costos.

Utilice la consola de ElastiCache, la AWS CLI o la API de ElastiCache para agregar, mostrar, modificar, eliminar o copiar las etiquetas de asignación de costos en sus copias de seguridad. Para obtener más información, consulte Monitoreo de costos con etiquetas de asignación de costos.

Tutorial: propagación de un nuevo clúster basado en nodos con una copia de seguridad creada externamente

Si crea un nuevo clúster basado en nodos de Valkey o Redis OSS, puede propagarlo con datos de un archivo de copia de seguridad .rdb de Valkey o Redis OSS. La propagación del clúster resulta útil si actualmente administra una instancia de Valkey o Redis OSS fuera de ElastiCache y desea rellenar el nuevo clúster basado en nodos de ElastiCache para Redis OSS con los datos de Valkey o Redis OSS existentes.

Para propagar un nuevo clúster basado en nodos de Valkey o Redis OSS desde una copia de seguridad de Valkey o Redis OSS creada en Amazon ElastiCache, consulte Restauración desde una copia de seguridad a una nueva caché.

Cuando use un archivo .rdb de Valkey o Redis OSS para propagar un nuevo clúster basado en nodos, podrá hacer lo siguiente:

- Actualizar desde un clúster no particionado a un clúster basado en nodos de Valkey o Redis OSS (modo de clúster habilitado) que ejecute la versión 3.2.4 de Redis OSS.
- Especificar un número de particiones (denominadas “grupos de nodos” en la API y la CLI) en el nuevo clúster basado en nodos. Este número puede ser distinto del número de particiones del clúster basado en nodos que se utilizó para crear el archivo de copia de seguridad.
- Especificar un tipo de nodo distinto para el nuevo clúster basado en nodos, más grande o más pequeño que el que se utilizó en el clúster que creó la copia de seguridad. Si escala a un tipo de nodo más pequeño, asegúrese de que el nuevo tipo de nodo tenga suficiente memoria para los datos y la sobrecarga de Valkey o Redis OSS. Para obtener más información, consulte Forma de garantizar que dispone de memoria suficiente para crear una instantánea de Valkey o Redis OSS.
- Distribuya las claves en las ranuras del nuevo clúster de Valkey o Redis OSS (modo de clúster habilitado) de manera distinta que en el clúster que se utilizó para crear el archivo de copia de seguridad.

> **Note**
>
> No puede inicializar un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) a partir de un archivo .rdb creado con un clúster de Valkey o Redis OSS (modo de clúster habilitado).

Important

- Debe asegurarse de que los datos de la copia de seguridad de Valkey o Redis OSS no superen los recursos del nodo. Por ejemplo, no puede cargar un archivo .rdb con 5 GB de datos de Valkey o Redis OSS a un nodo cache.m3.medium que tenga 2,9 GB de memoria.

  Si la copia de seguridad es demasiado grande, el clúster resultante tendrá el estado `restore-failed`. Si esto ocurre, deberá eliminar el clúster y empezar de nuevo.

  Para obtener una lista completa de las especificaciones y los tipos de nodos, consulte Parámetros específicos de tipos de nodos de Redis OSS y Características y detalles del producto de Amazon ElastiCache.
- Solo puede cifrar un archivo .rdb de Valkey o Redis OSS con cifrado del servidor de Amazon S3 (SSE-S3). Para obtener más información, consulte Protección de los datos con el cifrado del servidor.

A continuación, encontrará temas que lo guiarán por el proceso de migración de su clúster desde fuera de ElastiCache para Valkey o Redis OSS a ElastiCache para Redis OSS.

Migración a ElastiCache para Redis OSS

- Paso 1: crear una copia de seguridad de Valkey o Redis OSS
- Paso 2: crear un bucket y una carpeta de Amazon S3
- Paso 3: cargar la copia de seguridad a Amazon S3
- Paso 4: conceder acceso de lectura al archivo .rdb a ElastiCache

Migración desde servicios externos a ElastiCache para Redis OSS.

- Paso 1: crear una copia de seguridad de Valkey o Redis OSS
- Paso 2: crear un bucket y una carpeta de Amazon S3
- Paso 3: cargar la copia de seguridad a Amazon S3
- Paso 4: conceder acceso de lectura al archivo .rdb a ElastiCache

Paso 1: crear una copia de seguridad de Valkey o Redis OSS

Cómo crear la copia de seguridad de Valkey o Redis OSS para propagar la instancia de ElastiCache para Redis OSS

1. Conéctese a su instancia de Valkey o Redis OSS existente.
2. Ejecute la operación `BGSAVE` o `SAVE` para crear una copia de seguridad. Tenga en cuenta la ubicación de su archivo .rdb.

   `BGSAVE` es una operación asincrónica y no bloquea otros clientes durante el procesamiento. Para obtener más información, consulte la operación BGSAVE en el sitio web de Valkey.

   `SAVE` es una operación sincrónica y bloquea otros procesos hasta que finalice. Para obtener más información, consulte la operación SAVE en el sitio web de Valkey.

Para obtener más información adicional acerca de la creación de copias de seguridad, consulte Persistence en el sitio web de Valkey.

Paso 2: crear un bucket y una carpeta de Amazon S3

Una vez que se crea el archivo de copia de seguridad, deberá cargarlo en una carpeta de un bucket de Amazon S3. Para ello, primero debe disponer de un bucket de Amazon S3 y de una carpeta en dicho bucket. Si ya dispone de un bucket de Amazon S3 y una carpeta con los permisos pertinentes, puede pasar a Paso 3: cargar la copia de seguridad a Amazon S3.

Creación de un bucket de Amazon S3

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. Siga las instrucciones para crear un bucket de Amazon S3 en Creación de un bucket en la Guía del usuario de Amazon Simple Storage Service.

   El nombre del bucket de Amazon S3 debe estar conforme con DNS. De lo contrario, ElastiCache no podrá acceder al archivo de copia de seguridad. Las reglas para la conformidad con DNS son:

   - Los nombres deben tener un mínimo de 3 y un máximo de 63 caracteres de largo.
   - Los nombres deben ser una serie de una o más etiquetas separadas por un punto (.) en el que cada etiqueta:

- Comienza por una letra minúscula o un número.
- Termina con una letra minúscula o un número.
- Solo contiene letras minúsculas, números y guiones.

- Los nombres no pueden tener el formato de una dirección IP (por ejemplo, 192.0.2.0).

Debe crear su bucket de Amazon S3 en la misma región de AWS que su clúster de ElastiCache para Redis OSS nuevo. Este enfoque garantiza la mayor velocidad de transferencia de datos posible cuando ElastiCache lea el archivo .rdb desde Amazon S3.

> **Note**
>
> Para conservar la máxima seguridad de los datos, asegúrese de que los permisos de su bucket de Amazon S3 sean lo más restrictivos posible. Al mismo tiempo, los permisos seguirán teniendo que permitir que se utilicen el bucket y su contenido para inicializar su nuevo clúster de Valkey o Redis OSS.

Para agregar una carpeta a un bucket de Amazon S3

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. Elija el nombre del bucket en el que va a cargar el archivo .rdb.
3. Elija Crear carpeta.
4. Escriba un nombre para la nueva carpeta.
5. Seleccione Save.

   Anote el nombre del bucket y el nombre de la carpeta.

Paso 3: cargar la copia de seguridad a Amazon S3

Ahora, cargue el archivo .rdb que creó en Paso 1: crear una copia de seguridad de Valkey o Redis OSS. Cárguelo en el bucket de Amazon S3 y la carpeta que creó en Paso 2: crear un bucket y una carpeta de Amazon S3. Para obtener más información sobre esta tarea, consulte Agregar un objeto a un bucket. Entre los pasos 2 y 3, elija el nombre de la carpeta que creó.

Para cargar el archivo .rdb a una carpeta de Amazon S3

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. Elija el nombre del bucket de Amazon S3 que creó en el paso 2.
3. Elija el nombre de la carpeta que creó en el paso 2.
4. Seleccione Cargar.
5. Elija Add files.
6. Examine el archivo o los archivos que desea cargar y, a continuación, elija el archivo o los archivos. Para elegir varios archivos, mantenga pulsada la tecla Ctrl al mismo tiempo que selecciona un nombre de archivo.
7. Elija Open.
8. Asegúrese de que se muestran los archivos correctos en el cuadro de diálogo Upload y, a continuación, elija Upload.

Escriba la ruta del archivo .rdb. Por ejemplo, si el nombre del bucket es `myBucket` y la ruta es `myFolder/redis.rdb`, escriba `myBucket/myFolder/redis.rdb`. Necesitará esta ruta para propagar en el nuevo clúster los datos de la copia de seguridad.

Para obtener más información, consulte Restricciones y limitaciones de los buckets en la Guía del usuario de Amazon Simple Storage Service.

Paso 4: conceder acceso de lectura al archivo .rdb a ElastiCache

Ahora, conceda acceso de lectura a ElastiCache a su archivo de copia de seguridad .rdb. Puede conceder acceso a ElastiCache a su archivo de copia de seguridad de una manera distinta en función de si su bucket se encuentra en una región de AWS predeterminada o en una región de AWS con suscripción.

AWSLas regiones de que se presentaron antes del 20 de marzo de 2019 se encuentran habilitadas de forma predeterminada. Puede comenzar a trabajar en estas regiones de AWS de inmediato. Las regiones que se presentaron después del 20 de marzo de 2019, como Asia-Pacífico (Hong Kong) y Medio Oriente (Baréin), se encuentran deshabilitadas de forma predeterminada. Debe habilitar o suscribirse a estas regiones antes de poder utilizarlas, tal y como se describe en Administración de regiones de AWS en Referencia general de AWS.

Elija su enfoque en función de su región de AWS:

- Para una región predeterminada, utilice el procedimiento de Conceder acceso de lectura a ElastiCache al archivo .rdb en una región predeterminada.
- Para una región con suscripción, utilice el procedimiento de Otorgar acceso de lectura de ElastiCache al archivo .rdb en una región con suscripción.

Conceder acceso de lectura a ElastiCache al archivo .rdb en una región predeterminada

AWSLas regiones de que se presentaron antes del 20 de marzo de 2019 se encuentran habilitadas de forma predeterminada. Puede comenzar a trabajar en estas regiones de AWS de inmediato. Las regiones que se presentaron después del 20 de marzo de 2019, como Asia-Pacífico (Hong Kong) y Medio Oriente (Baréin), se encuentran deshabilitadas de forma predeterminada. Debe habilitar o suscribirse a estas regiones antes de poder utilizarlas, tal y como se describe en Administración de regiones de AWS en Referencia general de AWS.

Para conceder acceso de lectura a ElastiCache al archivo de copia de seguridad en una región de AWS habilitada de forma predeterminada

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. Elija el nombre del bucket de S3 que contiene su archivo .rdb.
3. Elija el nombre de la carpeta que contiene su archivo .rdb.
4. Elija el nombre de su archivo de copia de seguridad .rdb. El nombre del archivo seleccionado aparecerá encima de las pestañas, en la parte superior de la página.
5. Elija Permisos.
6. Si aws-scs-s3-readonly o uno de las identificaciones canónicas de la lista siguiente no aparece como usuario, haga lo siguiente:

    a. En Access for other AWS accounts (Acceso para otras cuentas de), elija Add grantee (Agregar beneficiario).

    b. En el cuadro, agregue el ID canónico de la región de AWS, tal como se muestra a continuación:

    - AWSRegión GovCloud (EE. UU. Oeste):

```
40fa568277ad703bd160f66ae4f83fc9dfdfd06c2f1b5060ca22442ac3ef8be6
```

> **Important**
>
> La copia de seguridad debe encontrarse en un bucket de S3, en AWS GovCloud (US), para que pueda descargarla a un clúster de Valkey o Redis OSS de AWS GovCloud (US).

- AWSRegiones de habilitadas de forma predeterminada

```
540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353
```

c. Establezca los permisos en el bucket eligiendo Yes (Sí) para los valores siguientes:

- Objetos List (Lista) o Write (Escribir)
- Read/write object ACL permissions (Permisos de lectura y escritura de ACL)

d. Seleccione Save.

7. Elija Overview (Información general) y, a continuación, elija Download (Descargar).

Otorgar acceso de lectura de ElastiCache al archivo .rdb en una región con suscripción

AWSLas regiones de que se presentaron antes del 20 de marzo de 2019 se encuentran habilitadas de forma predeterminada. Puede comenzar a trabajar en estas regiones de AWS de inmediato. Las regiones que se presentaron después del 20 de marzo de 2019, como Asia-Pacífico (Hong Kong) y Medio Oriente (Baréin), se encuentran deshabilitadas de forma predeterminada. Debe habilitar o suscribirse a estas regiones antes de poder utilizarlas, tal y como se describe en Administración de regiones de AWS en Referencia general de AWS.

Ahora, conceda acceso de lectura a ElastiCache a su archivo de copia de seguridad .rdb.

Para conceder a ElastiCache acceso de lectura al archivo de copia de seguridad

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. Elija el nombre del bucket de S3 que contiene su archivo .rdb.
3. Elija el nombre de la carpeta que contiene su archivo .rdb.
4. Elija el nombre de su archivo de copia de seguridad .rdb. El nombre del archivo seleccionado aparecerá encima de las pestañas, en la parte superior de la página.
5. Elija la pestaña Permisos.

6. En Permissions (Permisos), elija Bucket policy (Política de bucket), y luego Edit (Editar).
7. Actualice la política para conceder a ElastiCache los permisos necesarios a fin de realizar operaciones:

   - Agregue `[ "Service" : "region-full-name.elasticache-snapshot.amazonaws.com" ]` a `Principal`.
   - Agregue los siguientes permisos necesarios para exportar una instantánea al bucket de Amazon S3:
     - `"s3:GetObject"`
     - `"s3:ListBucket"`
     - `"s3:GetBucketAcl"`

   A continuación, se muestra un ejemplo del aspecto que tendría la política actualizada.

   JSON

```
{
    "Version":"2012-10-17",
    "Id": "Policy15397346",
    "Statement": [
        {
            "Sid": "Stmt15399483",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:ListBucket",
                "s3:GetBucketAcl"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/backup1.rdb",
                "arn:aws:s3:::amzn-s3-demo-bucket/backup2.rdb"
            ]
        }
    ]
}
```

8. Seleccione Save changes (Guardar cambios).

Inicialización del clúster de ElastiCache con los datos de archivos .rdb

Ahora se encuentra listo para crear un clúster de ElastiCache e inicializar los datos del archivo .rdb. Para crear el clúster, siga las instrucciones que se detallan en Creación de un clúster para Valkey o Redis OSS o Creación de un grupo de replicación de Valkey o Redis OSS desde cero. Asegúrese de elegir Valkey o Redis OSS como motor del clúster.

El método que utilice para indicar a ElastiCache dónde buscar la copia de seguridad que cargó en Amazon S3 dependerá del método que utilice para crear el clúster:

Propagación del clúster o grupo de replicación de ElastiCache para Redis OSS con los datos del archivo .rdb

- Uso de la consola de ElastiCache

  Al seleccionar la Cluster settings (Configuración del clúster), elija Restore from backups (Restaurar a partir de copias de seguridad) como método de creación del clúster y, a continuación, elija Other backups (Otras copias de seguridad) como Source (Origen) en la sección Backup source (Origen de copias de seguridad). En el cuadro Seed RDB file S3 location (Inicializar ubicación de S3 del archivo RDB), escriba la ruta de Amazon S3 de los archivos. Si tiene varios archivos.rdb, escriba la ruta para cada archivo en una lista separada por comas. La ruta de Amazon S3 tendrá un aspecto similar a *myBucket*/*myFolder*/*myBackupFilename*.rdb.

- Uso de AWS CLI

  Si usa la operación `create-cache-cluster` o `create-replication-group`, use el parámetro `--snapshot-arns` para especificar un ARN completo para cada archivo .rdb. Por ejemplo, `arn:aws:s3:::`*myBucket*/*myFolder*/*myBackupFilename*`.rdb`. El ARN debe resolverse en los archivos de copia de seguridad que almacenó en Amazon S3.

- Uso de la API de ElastiCache

  Si utiliza la operación de API de ElastiCache `CreateCacheCluster` o `CreateReplicationGroup`, utilice el parámetro `SnapshotArns` a fin de especificar un ARN completo para cada archivo .rdb. Por ejemplo, `arn:aws:s3:::`*myBucket*/*myFolder*/*myBackupFilename*`.rdb`. El ARN debe resolverse en los archivos de copia de seguridad que almacenó en Amazon S3.

> **Important**
>
> Al inicializar un clúster de Valkey o Redis OSS (modo de clúster habilitado), debe configurar cada grupo de nodos (partición) del nuevo clúster o grupo de replicación. Para ello, utilice el parámetro `--node-group-configuration` (API: `NodeGroupConfiguration`). Para obtener más información, consulte los siguientes temas:
>
> - CLI: create-replication-group en la Referencia de la AWS CLI
> - API: CreateReplicationGroup en la Referencia de la API de ElastiCache

Durante el proceso de creación de su clúster, los datos de su copia de seguridad de Valkey o Redis OSS se escribirán en el clúster. Puede monitorear el progreso al visualizar los mensajes de eventos de ElastiCache. Para ello, diríjase a la consola de ElastiCache y elija Cache Events (Eventos de caché). También puede utilizar la interfaz de línea de comandos de AWS ElastiCache o la API de ElastiCache para obtener los mensajes de eventos. Para obtener más información, consulte Visualización de eventos de ElastiCache.

Versiones del motor y actualización en ElastiCache

En esta sección, se detallan los motores admitidos de Valkey, Memcached y Redis OSS y se explica cómo actualizarlos. Tenga en cuenta que todas las características de Redis OSS 7.2 están disponibles en Valkey 7.2 y versiones posteriores de forma predeterminada. También puede actualizar algunos motores existentes de ElastiCache para Redis OSS a un motor de Valkey.

Actualización de las versiones del motor, incluidas las actualizaciones de varios motores

Valkey y Redis OSS

Con Valkey y Redis OSS, las actualizaciones de versiones de su clúster o grupo de replicación se inician al realizar modificaciones con la consola de ElastiCache, la AWS CLI o la API de ElastiCache y al especificar una versión del motor más reciente.

También puede realizar una actualización de varias versiones de Redis OSS a Valkey. Para obtener más información sobre la actualización de varias versiones, consulte Cómo actualizar de Redis OSS a Valkey.

Temas

- Cómo actualizar de Redis OSS a Valkey
- Resolución de actualizaciones bloqueadas del motor de Valkey o Redis OSS

Cómo modificar clústeres y grupos de reproducción	
Cachés	Grupos de reproducción
Uso de la Consola de administración de AWS de ElastiCache	Uso de Consola de administración de AWS
Uso de la AWS CLI con ElastiCache	Uso de AWS CLI
Uso de la API de ElastiCache	Uso de la API de ElastiCache

Memcached

Con Memcached, para comenzar las actualizaciones de versiones en su clúster, modifíquelo y especifique una versión del motor más reciente. Para ello, puede utilizar la consola de ElastiCache, la AWS CLI o la API de ElastiCache:

- Para utilizar la Consola de administración de AWS, consulte: Uso de la Consola de administración de AWS de ElastiCache.
- Para utilizar la AWS CLI, consulte Uso de la AWS CLI con ElastiCache.
- Para utilizar la API de ElastiCache, consulte Uso de la API de ElastiCache.

Cómo actualizar de Redis OSS a Valkey

Valkey está diseñado como un sustituto directo de Redis OSS 7. Puede actualizar de Redis OSS a Valkey mediante la consola, la API o la CLI, especificando el nuevo motor y la versión principal del motor. La actualización no modificará la dirección IP del punto de conexión ni ningún otro aspecto de la aplicación. Al actualizar desde Redis OSS 5.0.6 y versiones posteriores, no experimentará ningún tiempo de inactividad.

> **Note**
>
> Requisitos de la versión de la CLI de AWS para las actualizaciones de Redis OSS a Valkey:
>
> - Para la versión 1 de la CLI de AWS: versión mínima requerida 1.35.2 (versión actual: 1.40.22)
> - Para la versión 2 de la CLI de AWS: versión mínima requerida 2.18.2 (versión actual: 2.27.22)

> **Note**
>
> - Al actualizar desde una versión anterior a la 5.0.6 de Redis OSS, es posible que se produzca un tiempo de conmutación por error de 30 a 60 segundos durante la propagación del DNS.
> - Si desea actualizar un clúster de un solo nodo de Redis OSS (modo de clúster deshabilitado) existente al motor de Valkey, primero debe seguir estos pasos: Creación de un grupo de reproducción mediante un clúster existente. Después de añadir el clúster de

un solo nodo de Redis OSS (modo de clúster deshabilitado) en un grupo de replicación, puede actualizar desde varios motores a Valkey.

Actualización de un grupo de replicación de Redis OSS a Valkey

Si ya tiene un grupo de replicación de Redis OSS que está utilizando el grupo de parámetros de caché predeterminado, puede actualizarlo a Valkey especificando el nuevo motor y la versión del motor con la API modify-replication-group.

Para Linux, macOS o Unix:

```
aws elasticache modify-replication-group \
   --replication-group-id myReplGroup \
   --engine valkey \
   --engine-version 8.0
```

Para Windows:

```
aws elasticache modify-replication-group ^
   --replication-group-id myReplGroup ^
   --engine valkey ^
   --engine-version 8.0
```

Si tiene un grupo de parámetros de caché personalizado aplicado al grupo de replicación de Redis OSS existente que desea actualizar, también tendrá que incluir un grupo de parámetros de caché de Valkey personalizado en la solicitud. El grupo de parámetros personalizados de Valkey introducido debe tener los mismos valores de parámetros estáticos de Redis OSS que el grupo de parámetros personalizados de Redis OSS existente.

Para Linux, macOS o Unix:

```
aws elasticache modify-replication-group \
   --replication-group-id myReplGroup \
   --engine valkey \
   --engine-version 8.0 \
   --cache-parameter-group-name myParamGroup
```

Para Windows:

```
aws elasticache modify-replication-group ^
   --replication-group-id myReplGroup ^
   --engine valkey ^
   --engine-version 8.0 ^
   --cache-parameter-group-name myParamGroup
```

Actualización de una caché sin servidor de Redis OSS a Valkey con la CLI

Para Linux, macOS o Unix:

```
aws elasticache modify-serverless-cache \
   --serverless-cache-name myCluster \
   --engine valkey \
   --major-engine-version 8
```

Para Windows:

```
aws elasticache modify-serverless-cache ^
   --serverless-cache-name myCluster ^
   --engine valkey ^
   --major-engine-version 8
```

Actualización de Redis OSS a Valkey con la consola

Actualización desde Redis OSS 5 a Valkey

1. Seleccione la caché de Redis OSS que desea actualizar.
2. Debería aparecer la ventana Actualice a Valkey. Seleccione el botón Actualice a Valkey.
3. Vaya a Configuración de la caché y, a continuación, seleccione Versión del motor. Se recomienda utilizar la versión más reciente de Valkey.
4. Si esta caché es sin servidor, tendrá que actualizar el grupo de parámetros. Vaya al área Grupos de parámetros de la Configuración de la caché y seleccione un grupo de parámetros adecuado, como default.valkey8.
5. Seleccione Actualizar.

Esta caché ahora aparecerá en el área de Valkey de la consola.

> **Note**
>
> La actualización directa desde Redis OSS 4 o una versión anterior a Valkey puede incluir un tiempo de conmutación por error mayor, de 30 a 60 segundos, durante la propagación del DNS.

Cómo cambiar a una versión menor desde Valkey a Redis OSS

Si por alguna razón desea revertir la actualización del clúster, Amazon ElastiCache admite la reversión de una caché de Valkey 7.2 a Redis OSS 7.1. Puede revertir siguiendo los mismos pasos de la consola, la API o la CLI que en una actualización del motor y especificando Redis OSS 7.1 como la versión del motor de destino. Las reversiones utilizan los mismos procesos que los de una actualización. La reversión no modificará la dirección IP del punto de conexión ni todos los demás aspectos de la aplicación y tampoco se generará tiempo de inactividad.

Además, puede restaurar una instantánea creada a partir de la caché de Valkey 7.2 como una caché de Redis OSS 7.1. Al restaurar a partir de una instantánea, puede especificar Redis OSS 7.1 como la versión del motor de destino. Al utilizar esta opción, se creará una nueva caché a partir de la instantánea. La restauración a partir de una instantánea no afecta a la caché de Valkey desde la que se creó la instantánea.

Al revertir, se aplican los siguientes requisitos y limitaciones:

- ElastiCache solo admite la reversión desde Valkey 7.2 a Redis OSS 7.1. Esto es cierto incluso si ha actualizado a Valkey 7.2 desde una versión anterior a Redis OSS 7.1.
- Todos los grupos de usuarios y usuarios asociados al grupo de replicación o a la caché sin servidor que se va a revertir deben configurarse con el tipo de motor `REDIS`.

Resolución de actualizaciones bloqueadas del motor de Valkey o Redis OSS

Tal y como se muestra en la siguiente tabla, la operación de actualización del motor de Valkey o Redis OSS se bloqueará si tiene una operación de escalado vertical pendiente.

Operaciones pendientes	Operaciones bloqueadas
Escalado ascendente	Actualización del motor inmediata

Operaciones pendientes	Operaciones bloqueadas
Actualización del motor	Ampliación inmediata
Ampliación y actualización del motor	Ampliación inmediata
	Actualización del motor inmediata

Cómo resolver una actualización del motor de Valkey o Redis OSS bloqueada

- Realice una de las siguientes acciones:
  - Programe la operación de actualización del motor de Redis OSS o Valkey para el siguiente periodo de mantenimiento desactivando la casilla de verificación Aplicar inmediatamente.

    Con la CLI, utilice `--no-apply-immediately`. Con la API, utilice `ApplyImmediately=false`.
  - Espere al siguiente periodo de mantenimiento (o después) para realizar la operación de actualización del motor de Redis OSS.
  - Añada la operación de escalado vertical de Redis OSS a esta modificación de clúster con la casilla de verificación Aplicar inmediatamente activada.

    Con la CLI, utilice `--apply-immediately`. Con la API, utilice `ApplyImmediately=true`.

    Con este enfoque, la actualización del motor se realizará inmediatamente, por lo que quedará cancelada para el siguiente periodo de mantenimiento.

Soporte extendido de ElastiCache

Con el soporte extendido de ElastiCache, puede seguir ejecutando su caché en una versión principal del motor después de la fecha de finalización del soporte estándar por un precio adicional. Si no actualiza después de la fecha de finalización del soporte estándar, se le cobrará.

El soporte extendido ofrece las siguientes actualizaciones y soporte técnico:

- Actualizaciones de seguridad para las CVE críticas y elevadas para la caché y el motor de la caché
- Correcciones de errores y parches para problemas críticos

- Posibilidad de abrir casos de soporte y recibir ayuda para la solución de problemas según el acuerdo de nivel de servicio de ElastiCache

Esta oferta de pago le da más tiempo para llevar a cabo la actualización a una versión principal del motor compatible.

Por ejemplo, la fecha de finalización del soporte estándar de ElastiCache para Redis OSS 4.0.10 es el 31 de enero de 2026. Si no se ha preparado para actualizar manualmente a Valkey o a Redis OSS 6 o posterior antes de esa fecha, ElastiCache inscribirá automáticamente sus cachés en el soporte extendido y podrá seguir ejecutando Redis OSS 4.0.10. A partir del primer día del mes siguiente a la finalización del soporte estándar, el 1 de febrero de 2026, ElastiCache le cobrará automáticamente por el soporte extendido.

El soporte extendido está disponible durante un máximo de 3 años después de la fecha de finalización del soporte estándar para una versión principal del motor. En el caso de las versiones 4 y 5 de ElastiCache para Redis OSS, será el 31 de enero de 2029. Después de esta fecha, todas las cachés que sigan ejecutando las versiones 4 y 5 de Redis OSS se actualizarán automáticamente a la última versión de Valkey.

Cuando finalice el período de soporte de un motor, las cachés que sigan ejecutando esa versión anterior pasarán automáticamente al soporte extendido. Se le notificará antes de la fecha de inicio de los precios de soporte extendido para que, en su lugar, pueda actualizar su instancia. También puede renunciar de forma explícita en cualquier momento actualizándola a las versiones compatibles.

Para obtener más información sobre las fechas de finalización del soporte estándar y las fechas de finalización del soporte extendido, consulte Programa de fin de vida útil de las versiones de ElastiCache para Redis OSS para Valkey, Memcached o Redis OSS.

Temas

- Precios del soporte extendido de ElastiCache
- Versiones con el soporte extendido de ElastiCache
- ElastiCache y las responsabilidades del cliente con el soporte extendido de ElastiCache

Precios del soporte extendido de ElastiCache

Todos los motores inscritos en el soporte extendido de ElastiCache incurrirán en cargos a partir del día siguiente a la fecha de finalización del soporte estándar. Para conocer la fecha de finalización del soporte estándar de ElastiCache, consulte Versiones con el soporte extendido de ElastiCache.

El cargo adicional del soporte extendido de ElastiCache se detendrá automáticamente si realiza una de las siguientes acciones:

- Actualizar a una versión de motor incluida en el soporte estándar.
- Eliminar la caché en la que se ejecuta una versión principal pasada la fecha de finalización del soporte estándar de ElastiCache.

Los cargos se reiniciarán si la versión del motor de destino se incluye en el soporte extendido de ElastiCache en el futuro.

Por ejemplo, supongamos que la versión 4 de ElastiCache para Redis OSS entra en el soporte extendido el 1 de febrero de 2026 y que usted actualiza las cachés de la versión 4 a la versión 6 el 1 de enero de 2027. Solo se le cobrarán 11 meses de soporte extendido de la versión 4 de ElastiCache para Redis OSS. Si continúa ejecutando la versión 6 de ElastiCache para Redis OSS después de la fecha de finalización del soporte estándar, que es el 31 de enero de 2027, esas cachés volverán a incurrir en cargos por el soporte extendido a partir del 1 de febrero de 2027.

Puede evitar que se le cobre por el soporte extendido de ElastiCache impidiendo que ElastiCache cree o restaure una caché pasada la fecha de finalización del soporte estándar de ElastiCache.

Para obtener más información, consulte Precios de Amazon ElastiCache.

Versiones con el soporte extendido de ElastiCache

Las versiones 4 y 5 del software de código abierto (OSS) de Redis llegaron al final de su vida útil en la comunidad en 2020 y 2022, respectivamente. Esto significa que la comunidad no lanzará más actualizaciones, correcciones de errores ni parches de seguridad. El soporte estándar para las versiones 4 y 5 de ElastiCache Redis OSS en ElastiCache finalizará el 31 de enero de 2026. Si continúa utilizando versiones no compatibles de Redis OSS, sus datos podrían quedar expuestos a Vulnerabilidades y exposiciones comunes (CVE) conocidas.

A partir del 1 de febrero de 2026, las cachés de ElastiCache que aún se estén ejecutando en las versiones 4 y 5 de Redis OSS se inscribirán automáticamente en el soporte extendido para ofrecer disponibilidad y seguridad continuas. Si bien el soporte extendido ofrece flexibilidad, recomendamos

considerar el final del soporte estándar como un hito en la planificación de sus cargas de trabajo de producción. Le recomendamos encarecidamente que actualice sus cachés de Redis OSS versión 4 y 5 a ElastiCache para Valkey o Redis OSS versión 6 o posterior, antes de que finalice el soporte estándar.

En la siguiente tabla se muestra la fecha de finalización del soporte estándar de Amazon ElastiCache y las fechas del soporte extendido.

Programación del soporte extendido y del fin de la vida útil

Versión principal del motor	Fecha de finalización del soporte estándar	Inicio del soporte extendido Y1 Premium	Inicio del soporte extendido Y2 Premium	Inicio del soporte extendido Y3 Premium	Fin del soporte extendido y versión EOL
Redis OSS versión 4	31/1/2026	1/2/2026	1/2/2027	1/2/2028	31/1/2029
Redis OSS versión 5	31/1/2026	1/2/2026	1/2/2027	1/2/2028	31/1/2029
Redis OSS versión 6	31/1/2027	1/2/2027	1/2/2028	1/2/2029	31/1/2030

El soporte extendido solo se ofrecerá para la última versión de parche compatible de cada una de las versiones principales de Redis OSS. Si el soporte extendido comienza el 1 de febrero de 2026, si los clústeres de Redis OSS versión 4 y 5 aún no tienen las versiones más recientes del parche, se actualizarán automáticamente a la versión 4.0.10 para Redis OSS versión 4 y a la versión 5.0.6 para Redis OSS versión 5 antes de inscribirse en el soporte extendido. De este modo se garantiza que reciba las actualizaciones de seguridad y las correcciones de errores a través del soporte extendido. No necesita realizar ninguna acción para actualizar a las versiones más recientes de los parches como parte de la transición al soporte extendido.

ElastiCache y las responsabilidades del cliente con el soporte extendido de ElastiCache

A continuación se detallan las responsabilidades de Amazon ElastiCache y sus responsabilidades con el soporte extendido de ElastiCache.

Responsabilidades de Amazon ElastiCache

Tras la fecha de finalización del soporte estándar de ElastiCache, Amazon ElastiCache proporcionará parches, correcciones de errores y actualizaciones para los motores inscritos en el soporte extendido de ElastiCache. Esto ocurrirá durante un máximo de 3 años o hasta que deje de utilizar el soporte extendido de los motores, lo que ocurra primero.

Sus responsabilidades

Usted debe aplicar los parches, las correcciones de errores y las actualizaciones proporcionadas para las cachés en el soporte extendido de ElastiCache. Amazon ElastiCache se reserva el derecho de modificar, sustituir o retirar dichos parches, correcciones de errores y actualizaciones en cualquier momento. Si se necesita un parche para solucionar problemas de seguridad o estabilidad críticos, Amazon ElastiCache se reserva el derecho de actualizar sus cachés con el parche o bien de solicitarle que instale el parche.

También es responsable de actualizar el motor a una versión más reciente antes de la fecha de finalización del soporte extendido de ElastiCache. La fecha de finalización del soporte extendido de ElastiCache suele ser 3 años después de la fecha de finalización del soporte estándar de ElastiCache.

Si no actualiza el motor, después de la fecha de finalización del soporte extendido de ElastiCache, Amazon ElastiCache intentará actualizar el motor a una versión más reciente del motor que admita el soporte estándar de ElastiCache. Si se produce un error en la actualización, Amazon ElastiCache se reserva el derecho de eliminar la memoria caché en la que se ejecuta el motor una vez pasada la fecha de finalización del soporte estándar de ElastiCache. Sin embargo, antes de hacerlo, Amazon ElastiCache conservará los datos de ese motor.

Administración de versiones para ElastiCache

Administre la forma en que desea actualizar las cachés de ElastiCache y los clústeres basados en nodos actualizados para los motores de Valkey, Memcached y Redis OSS.

Administración de versiones para la caché sin servidor de ElastiCache

Controle si se realizan actualizaciones de la caché de ElastiCache Serverless, así como cuándo se realizan estas actualizaciones, y lleve a cabo las actualizaciones a su manera y cuando usted lo decida.

ElastiCache sin servidor aplica automáticamente la última versión del software secundario y de parche a la memoria caché, sin que la aplicación sufra ningún tiempo de inactividad. No tiene que hacer nada.

Cuando haya una nueva versión principal disponible, ElastiCache sin servidor le enviará una notificación en la consola y un evento en EventBridge. Puede optar por actualizar la memoria caché a la última versión principal modificando la memoria caché mediante la consola, la CLI o la API y seleccionando la versión más reciente del motor. De igual modo que en las actualizaciones menores y de parches, las actualizaciones de las versiones principales se realizan sin que se produzca ningún tiempo de inactividad de la aplicación.

Administración de versiones para clústeres de ElastiCache basados en nodos

Al trabajar con clústeres de ElastiCache basados en nodos, puede controlar el momento de la actualización del software de su clúster a nuevas versiones compatibles con ElastiCache. Puede controlar cuándo actualizar la caché a las versiones principal, secundaria y de parche más recientes disponibles. Para iniciar la actualización de las versiones del motor en el clúster o el grupo de reproducción, modifíquelo y especifique una nueva versión del motor.

Puede controlar la actualización del software compatible con los protocolos de su clúster a nuevas versiones compatibles con ElastiCache, así como el momento en que se realizan dichas actualizaciones. Este nivel de control permite mantener la compatibilidad con versiones concretas, probar nuevas versiones con la aplicación antes de implementarlas en producción y realizar actualizaciones de versiones en los horarios y los plazos que más le convengan.

Como las actualizaciones de versión pueden conllevar algunos riesgos de compatibilidad, no se producen automáticamente. Debe iniciarlas.

Clústeres de Valkey y Redis OSS

> **Note**
>
> - Si un clúster de Valkey o Redis OSS se replica en una o más regiones, la versión del motor se actualiza para las regiones secundarias y, después, para la región principal.
> - Las versiones de ElastiCache para Redis OSS se identifican con una versión semántica que incluye un componente principal y uno secundario. Por ejemplo, en Redis OSS 6.2, la versión principal es 6 y la versión secundaria es 2. Cuando se utilizan clústeres basados en nodos, ElastiCache para Redis OSS también expone el componente de parche; por ejemplo, en Redis OSS 6.2.1, la versión del parche es 1.

> Las versiones principales son para cambios incompatibles con la API y las versiones secundarias son para nuevas funciones que se añaden de manera compatible con versiones anteriores. Las versiones de parche son para correcciones de errores compatibles con versiones anteriores y cambios no funcionales.

Con Valkey y Redis OSS, para iniciar la actualización de las versiones del motor en el clúster o el grupo de replicación, modifíquelo y especifique una nueva versión del motor. Para obtener más información, consulte Modificación de un grupo de reproducción.

Memcached

Con Memcached, para actualizar a una versión más reciente, debe modificar su cluster y especificar la nueva versión del motor que desea utilizar. La actualización a una nueva versión de Memcached es un proceso destructivo: perderá los datos y deberá comenzar con una caché nueva. Para obtener más información, consulte Modificación de un clúster de ElastiCache.

Debe tener en cuenta los requisitos siguientes a la hora de actualizar de una versión antigua de Memcached a la versión 1.4.33 o posterior. Se produce un error con `CreateCacheCluster` y `ModifyCacheCluster` en las condiciones que se describen a continuación:

- Si `slab_chunk_max > max_item_size`.
- Si `max_item_size modulo slab_chunk_max != 0`.
- Si `max_item_size > ((max_cache_memory - memcached_connections_overhead) / 4)`.

  El valor `(max_cache_memory - memcached_connections_overhead)` es la memoria útil del nodo para los datos. Para obtener más información, consulte Capacidad adicional para conexiones de Memcached.

Motores y versiones compatibles

Las cachés sin servidor de ElastiCache admiten ElastiCache versión 7.2 para Valkey y versiones posteriores, ElastiCache versión 1.6 para Memcached y versiones posteriores y ElastiCache 7.0 para Redis OSS y versiones posteriores.

Los clústeres de ElastiCache basados en nodos admiten ElastiCache versión 7.2 para Valkey y versiones posteriores, ElastiCache versión 1.4.5 para Memcached y versiones posteriores y ElastiCache 4.0.10 para Redis OSS y versiones posteriores.

Los clústeres de ElastiCache basados en nodos son compatibles con las siguientes versiones de Valkey:

- Versiones de Valkey compatibles
- ElastiCache versión 8.2 para Valkey
- ElastiCache versión 8.1 para Valkey
- ElastiCache versión 8.0 para Valkey
- ElastiCache versión 7.2.6 para Valkey

Versiones de Valkey compatibles

Estas son las versiones de Valkey compatibles. Tenga en cuenta que Valkey es compatible con la mayoría de las características disponibles en ElastiCache versión 7.2 para Redis OSS de forma predeterminada.

- También puede actualizar los clústeres de ElastiCache con versiones anteriores a la 5.0.6. El proceso involucrado es el mismo, pero puede incurrir en un tiempo de conmutación por error más largo durante la propagación de DNS (de 30 s a 1 m).
- A partir de Redis OSS 7, ElastiCache admite el cambio entre Valkey o Redis OSS (modo de clúster deshabilitado) y Valkey o Redis OSS (modo de clúster habilitado).
- El proceso de actualización del motor de Amazon ElastiCache para Redis OSS se ha diseñado para hacer todo lo posible por retener los datos existentes y requiere que la replicación de Redis OSS se realice correctamente.
- Al actualizar el motor, ElastiCache cancelará las conexiones de cliente existentes. Para minimizar el tiempo de inactividad durante las actualizaciones del motor, le recomendamos que implemente las prácticas recomendadas para los clientes de Redis OSS, con reintentos de errores y retrocesos exponenciales, así como las prácticas recomendadas para minimizar el tiempo de inactividad durante el mantenimiento.
- No puede actualizar directamente de Valkey o Redis OSS (modo de clúster deshabilitado) a Valkey o Redis OSS (modo de clúster habilitado) cuando actualiza su motor. El siguiente procedimiento muestra cómo actualizar de Valkey o Redis OSS (modo de clúster deshabilitado) a Valkey o Redis OSS (modo de clúster habilitado).

Actualización de una versión de motor de Valkey o Redis OSS (modo de clúster deshabilitado) a la versión del motor de Valkey o Redis OSS (modo de clúster habilitado)

1. Realice una copia de seguridad de su clúster o grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado). Para obtener más información, consulte Copias de seguridad manuales.
2. Utilice la copia de seguridad para crear y propagar un clúster de Valkey o Redis OSS (modo de clúster habilitado) con una partición (grupo de nodo). Especifique la nueva versión de motor y habilite el modo de clúster al crear el clúster o grupo de reproducción. Para obtener más información, consulte Tutorial: propagación de un nuevo clúster basado en nodos con una copia de seguridad creada externamente.
3. Elimine el clúster o el grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) anterior. Para obtener más información, consulte Eliminación de un clúster en ElastiCache o Eliminación de un grupo de reproducción.
4. Escale el nuevo grupo de replicación o clúster de Valkey o Redis OSS (modo de clúster habilitado) al número de particiones (grupos de nodo) que necesita. Para obtener más información, consulte Escalado de clústeres de Valkey o Redis OSS (modo de clúster habilitado)

- Cuando actualiza las versiones principales del motor, por ejemplo de 5.0.6 a 6.0, debe seleccionar un grupo de parámetros nuevo que sea compatible con la versión del motor nueva.
- Para clústeres de Redis OSS sencillos y clústeres con multi-AZ deshabilitado, recomendamos disponer de suficiente memoria para Redis OSS, tal y como se describe en Forma de garantizar que dispone de memoria suficiente para crear una instantánea de Valkey o Redis OSS. En estos casos, el nodo principal no está disponible para las solicitudes de servicio durante el proceso de actualización.
- Para clústeres de Redis OSS con multi-AZ habilitado, también recomendamos que programe actualizaciones del motor durante los periodos de poco tráfico entrante. Cuando se actualiza a Redis OSS 5.0.6 o a una versión posterior, el clúster principal sigue disponible para atender solicitudes durante el proceso de actualización.

  Los clústeres y grupos de reproducción con varias particiones se procesan y se aplican parches de la siguiente manera:

  - Todas las particiones se procesan en paralelo. Solo se realiza una operación de actualización en una partición a la vez.

  - En cada partición, todas las réplicas se procesan antes que el principal. Si hay menos réplicas en una partición, el principal de esa partición podrá procesarse antes que las réplicas de otras particiones terminen de procesarse.
  - En todas las particiones, los nodos principales se procesan en series. Solo se actualiza un nodo principal a la vez.
- Si el cifrado se encuentra habilitado en su grupo de reproducción o clúster actual, no puede actualizar a una versión del motor que no admita cifrado, como de la versión 3.2.6 a la 3.2.10.

Consideraciones sobre Memcached

Cuando actualice un clúster de Memcached basado en nodos, tenga en cuenta lo siguiente.

- La administración de la versión del motor está diseñada para que pueda tener el mayor control posible sobre cómo se produce la aplicación de parches. Sin embargo, ElastiCache se reserva el derecho de aplicar parches en el clúster en su nombre, llegado el caso improbable de una vulnerabilidad de seguridad crítica en el sistema o el software de caché.
- Puesto que el motor de Memcached no es compatible con la persistencia, las actualizaciones de versión del motor de Memcached son siempre un proceso disruptivo que borra todos los datos de caché del clúster.

ElastiCache versión 8.2 para Valkey

Estas son algunas de las nuevas características que se han introducido en Valkey 8.2 (en comparación con ElastiCache Valkey 8.1):

- Compatibilidad nativa con la búsqueda vectorial, que le permite almacenar, indexar, buscar y actualizar miles de millones de incrustaciones vectoriales de alta dimensionalidad en la memoria con latencias tan bajas como de microsegundos.

Para obtener más información sobre Valkey, consulte Valkey.

Para obtener más información sobre la versión 8.2 de Valkey, que incluye la búsqueda vectorial, consulte Valkey Search.

ElastiCache versión 8.1 para Valkey

Estas son algunas de las nuevas características que se introdujeron en Valkey 8.1 (en comparación con ElastiCache Valkey 8.0):

- Una nueva implementación de tabla hash que reduce la sobrecarga de memoria para reducir el uso de memoria hasta un 20 % en el caso de los patrones de clave/valor más comunes.
- Compatibilidad nativa con los filtros de Bloom, un nuevo tipo de datos que permite realizar búsquedas utilizando hasta un 98 % menos de memoria en comparación con el tipo de datos Set.
- Nuevo comando COMMANDLOG que registra las ejecuciones lentas y las solicitudes y respuestas de gran tamaño.
- Nueva compatibilidad con la actualización condicional del comando SET con el argumento IFEQ.
- Mejoras del rendimiento, que incluyen una latencia hasta un 45 % menor para el comando ZRANK, un rendimiento hasta 12 veces más rápido para PFMERGE y PFCOUNT, y un rendimiento hasta un 514 % mayor para BITCOUNT.

Para obtener más información sobre Valkey, consulte Valkey

Para obtener más información sobre la versión 8.1 de Valkey, consulte las Valkey 8.1 Release Notes.

ElastiCache versión 8.0 para Valkey

Estas son algunas de las nuevas características que se han introducido en Valkey 8.0 (en comparación con ElastiCache Valkey 7.2.6):

- Mejoras en la eficiencia de la memoria, que permiten a los usuarios almacenar hasta un 20 % más de datos por nodo sin realizar cambios en la aplicación.
- Recientemente se ha introducido una infraestructura de métricas por ranura para clústeres basados en nodos, que proporciona una visibilidad detallada del rendimiento y el uso de recursos de las ranuras individuales.
- ElastiCache sin servidor para Valkey 8.0 puede duplicar las solicitudes por segundo (RPS) admitidas cada 2 o 3 minutos, alcanzando 5 millones de RPS por caché desde cero en menos de 13 minutos, con una latencia de lectura constante de p50 de menos de un milisegundo.

Para obtener más información sobre Valkey, consulte Valkey

Para obtener más información sobre la versión 8 de Valkey, consulte las Valkey 8 Release Notes.

ElastiCache versión 7.2.6 para Valkey

El 10 de octubre de 2024, se lanzó la versión 7.2.6 de ElastiCache para Valkey. Estas son algunas de las nuevas características que se introdujeron en la versión 7.2 (en comparación con ElastiCache versión 7.1 para Redis):

- Optimizaciones de rendimiento y memoria para varios tipos de datos: optimización de la memoria para teclas de tipo lista y conjunto, optimización de la velocidad para comandos de conjuntos ordenados, optimización del rendimiento para comandos con varias claves en modo clúster, mejoras del rendimiento de pub/sub, optimización del rendimiento para comandos SCAN, SSCAN, HSCAN, ZSCAN y muchas otras optimizaciones más pequeñas.
- Nueva opción WITHSCORE para los comandos ZRANK y ZREVRANK
- CLIENT NO-TOUCH permite a los clientes ejecutar comandos sin que eso afecte a la LRU/LFU de las claves.
- Nuevo comando CLUSTER MYSHARDID que devuelve el ID de partición del nodo para agrupar los nodos de forma lógica en el modo de clúster en función de la replicación.

Para obtener más información sobre Valkey, consulte Valkey

Para obtener más información sobre la versión 7.2 de ElastiCache para Valkey, consulte Redis OSS 7.2.4 Release Notes (ElastiCache versión 7.2 para Valkey incluye todos los cambios de ElastiCache versión 7.1 para Redis OSS hasta la versión 7.2.4 de ElastiCache para Redis OSS). Valkey 7.2 release notes de Valkey en GitHub.

ElastiCache versión 8.2 para Valkey

Estas son algunas de las nuevas características que se han introducido en Valkey 8.2 (en comparación con ElastiCache Valkey 8.1):

- Compatibilidad nativa con la búsqueda vectorial, que le permite almacenar, indexar, buscar y actualizar miles de millones de incrustaciones vectoriales de alta dimensionalidad en la memoria con latencias tan bajas como de microsegundos.

Para obtener más información sobre Valkey, consulte Valkey.

Para obtener más información sobre la versión 8.2 de Valkey, que incluye la búsqueda vectorial, consulte Valkey Search.

ElastiCache versión 8.1 para Valkey

Estas son algunas de las nuevas características que se introdujeron en Valkey 8.1 (en comparación con ElastiCache Valkey 8.0):

- Una nueva implementación de tabla hash que reduce la sobrecarga de memoria para reducir el uso de memoria hasta un 20 % en el caso de los patrones de clave/valor más comunes.
- Compatibilidad nativa con los filtros de Bloom, un nuevo tipo de datos que permite realizar búsquedas utilizando hasta un 98 % menos de memoria en comparación con el tipo de datos Set.
- Nuevo comando COMMANDLOG que registra las ejecuciones lentas y las solicitudes y respuestas de gran tamaño.
- Nueva compatibilidad con la actualización condicional del comando SET con el argumento IFEQ.
- Mejoras del rendimiento, que incluyen una latencia hasta un 45 % menor para el comando ZRANK, un rendimiento hasta 12 veces más rápido para PFMERGE y PFCOUNT, y un rendimiento hasta un 514 % mayor para BITCOUNT.

Para obtener más información sobre Valkey, consulte Valkey

Para obtener más información sobre la versión 8.1 de Valkey, consulte las Valkey 8.1 Release Notes.

ElastiCache versión 8.0 para Valkey

Estas son algunas de las nuevas características que se han introducido en Valkey 8.0 (en comparación con ElastiCache Valkey 7.2.6):

- Mejoras en la eficiencia de la memoria, que permiten a los usuarios almacenar hasta un 20 % más de datos por nodo sin realizar cambios en la aplicación.
- Recientemente se ha introducido una infraestructura de métricas por ranura para clústeres basados en nodos, que proporciona una visibilidad detallada del rendimiento y el uso de recursos de las ranuras individuales.
- ElastiCache sin servidor para Valkey 8.0 puede duplicar las solicitudes por segundo (RPS) admitidas cada 2 o 3 minutos, alcanzando 5 millones de RPS por caché desde cero en menos de 13 minutos, con una latencia de lectura constante de p50 de menos de un milisegundo.

Para obtener más información sobre Valkey, consulte Valkey

Para obtener más información sobre la versión 8 de Valkey, consulte las Valkey 8 Release Notes.

ElastiCache versión 7.2.6 para Valkey

El 10 de octubre de 2024, se lanzó la versión 7.2.6 de ElastiCache para Valkey. Estas son algunas de las nuevas características que se introdujeron en la versión 7.2 (en comparación con ElastiCache versión 7.1 para Redis):

- Optimizaciones de rendimiento y memoria para varios tipos de datos: optimización de la memoria para teclas de tipo lista y conjunto, optimización de la velocidad para comandos de conjuntos ordenados, optimización del rendimiento para comandos con varias claves en modo clúster, mejoras del rendimiento de pub/sub, optimización del rendimiento para comandos SCAN, SSCAN, HSCAN, ZSCAN y muchas otras optimizaciones más pequeñas.
- Nueva opción WITHSCORE para los comandos ZRANK y ZREVRANK
- CLIENT NO-TOUCH permite a los clientes ejecutar comandos sin que eso afecte a la LRU/LFU de las claves.
- Nuevo comando CLUSTER MYSHARDID que devuelve el ID de partición del nodo para agrupar los nodos de forma lógica en el modo de clúster en función de la replicación.

Para obtener más información sobre Valkey, consulte Valkey

Para obtener más información sobre la versión 7.2 de ElastiCache para Valkey, consulte Redis OSS 7.2.4 Release Notes (ElastiCache versión 7.2 para Valkey incluye todos los cambios de ElastiCache versión 7.1 para Redis OSS hasta la versión 7.2.4 de ElastiCache para Redis OSS). Valkey 7.2 release notes de Valkey en GitHub.

Versiones compatibles del motor de Redis OSS

Las cachés de ElastiCache sin servidor y los clústeres basados en nodos son compatibles con todas las versiones 7.1 y anteriores de Redis OSS.

- Versión 7.1 de ElastiCache para Redis OSS (mejorada)

Los clústeres de ElastiCache basados en nodos son compatibles con las siguientes versiones de Redis OSS:

- Versión 7.1 de ElastiCache para Redis OSS (mejorada)
- Versión 7.0 de ElastiCache para Redis OSS (mejorada)
- Versión 6.2 de ElastiCache para Redis OSS (mejorada)

- Versión 6.0 de ElastiCache para Redis OSS (mejorada)
- Versión 5.0.6 de ElastiCache para Redis OSS (mejorada)
- ElastiCache versión 5.0.5 para Redis OSS (obsoleta, use la versión 5.0.6)
- ElastiCache versión 5.0.4 para Redis OSS (obsoleta, use la versión 5.0.6)
- ElastiCache versión 5.0.3 para Redis OSS (obsoleta, use la versión 5.0.6)
- ElastiCache versión 5.0.0 para Redis OSS (obsoleta, use la versión 5.0.6)
- Versión 4.0.10 de ElastiCache para Redis OSS (mejorada)
- Versiones 3.x de fin de vida útil (EOL) anteriores
- Versiones 2.x de fin de vida útil (EOL) anteriores

Versión 7.1 de ElastiCache para Redis OSS (mejorada)

Esta versión contiene mejoras de rendimiento que permiten que las cargas de trabajo aumenten el rendimiento y reduzcan las latencias en las operaciones. La versión 7.1 de ElastiCache para Redis OSS presenta dos mejoras principales:

Hemos ampliado la funcionalidad mejorada de los subprocesos de E/S para gestionar también la lógica de la capa de presentación. Al hablar de la capa de presentación, nos referimos a los subprocesos de E/S mejorados; ahora, no solo leen la entrada del cliente, sino que también analizan la entrada en el formato de comando binario de Redis OSS. Luego, se reenvía al subproceso principal para su ejecución, lo que proporciona un aumento del rendimiento. Patrón mejorado de acceso a la memoria de Redis OSS. Los pasos de ejecución de muchas operaciones de estructura de datos están intercalados para garantizar el acceso paralelo a la memoria y reducir la latencia de acceso a esta. Al ejecutar ElastiCache en `R7g.4xlarge` Graviton3 o superior, los clientes pueden obtener más de un millón de solicitudes por segundo y por nodo. Con las mejoras de rendimiento de ElastiCache para Redis OSS versión 7.1, los clientes pueden lograr hasta un 100 % más de rendimiento y un 50 % menos de latencia P99 en comparación con ElastiCache para Redis OSS versión 7.0. Estas mejoras están disponibles en tamaños de nodo con al menos ocho núcleos físicos (`2xlarge` en Graviton y `4xlarge` en x86), independientemente del tipo de CPU, y no es necesario hacer cambios en el cliente.

> **Note**
>
> ElastiCache 7.1 es compatible con Redis OSS 7.0.

Versión 7.0 de ElastiCache para Redis OSS (mejorada)

ElastiCache para Redis OSS 7.0 añade una serie de mejoras y compatibilidad con nuevas funciones:

- Funciones: ElastiCache para Redis OSS 7 añade compatibilidad para funciones de Redis OSS y proporciona una experiencia administrada que permite a los desarrolladores ejecutar scripts de LUA con la lógica de la aplicación almacenada en el clúster de ElastiCache, sin necesidad de que los clientes vuelvan a enviar los scripts al servidor con cada conexión.
- Mejoras de ACL: Valkey y Redis OSS 7 añaden compatibilidad con la próxima versión de las listas de control de acceso (ACL). Los clientes ahora pueden especificar varios conjuntos de permisos en claves o espacios de claves específicos en Valkey y Redis OSS.
- Pub/Sub particionadas: ElastiCache para Valkey y Redis OSS 7 añade compatibilidad para ejecutar la funcionalidad Pub/Sub de forma particionada cuando se ejecuta ElastiCache en el Modo de clúster habilitado (CME). Las capacidades de publicación/suscripción permiten a los editores emitir mensajes a cualquier número de suscriptores de un canal. Los canales se enlazan a una partición del clúster de ElastiCache, lo que elimina la necesidad de propagar la información del canal entre las particiones, lo que mejora la escalabilidad.
- Multiplexación de E/S mejorada: ElastiCache para Valkey y Redis OSS 7 introduce la multiplexación de E/S mejorada, que ofrece un mayor rendimiento y una latencia reducida para cargas de trabajo de alto rendimiento que tienen muchas conexiones de cliente simultáneas a un clúster de ElastiCache. Por ejemplo, al utilizar un clúster de nodos r6g.xlarge y ejecutar 5200 clientes simultáneos, puede lograr un aumento de hasta un 72 % en el rendimiento (operaciones de lectura y escritura por segundo) y una disminución de la latencia de P99 de hasta un 71 %, en comparación con la versión 6 de ElastiCache para Redis OSS.

Para obtener más información sobre Valkey, consulte Valkey. Para obtener más información sobre la versión 7.0 de Redis OSS, consulte Redis OSS 7.0 Release Notes en Redis OSS en GitHub.

Versión 6.2 de ElastiCache para Redis OSS (mejorada)

ElastiCache para Redis OSS 6.2 incluye mejoras de rendimiento para clústeres habilitados para TLS que utilizan tipos de nodos x86 con 8 vCPUs o más o tipos de nodos Graviton2 con 4 vCPUs o más. Estas mejoras optimizan el rendimiento y reducen el tiempo de establecimiento de la conexión de clientes mediante la descarga del cifrado a otras vCPUs. Con Redis OSS 6.2, también puede administrar el acceso a los canales de publicación/suscripción con reglas de Lista de control de acceso (ACL).

Con esta versión, también presentamos soporte para la organización de datos en niveles en nodos de clúster que contienen SSD NVMe adjunta localmente. Para obtener más información, consulte Organización de datos en niveles en ElastiCache.

La versión 6.2.6 del motor de Redis OSS también introduce compatibilidad con el formato JavaScript Object Notation (JSON) nativo, una forma sencilla y sin esquemas de codificar conjuntos de datos complejos dentro de clústeres de Redis OSS. Con la compatibilidad con JSON, puede aprovechar el rendimiento y las API de Redis OSS para las aplicaciones que funcionan sobre JSON. Para obtener más información, consulte Introducción a JSON. También se incluyen métricas relacionadas con JSON, `JsonBasedCmds` y `JsonBasedCmdsLatency`, que se incorporan a CloudWatch para supervisar el uso de este tipo de datos. Para obtener más información, consulte Métricas de Valkey y Redis OSS.

Para especificar la versión del motor, utilice 6.2. ElastiCache invocará automáticamente la versión del parche preferida de Redis OSS 6.2 que se encuentre disponible. Por ejemplo, al crear o modificar un clúster, establece el parámetro `--engine-version` en 6.2. El clúster se lanzará con la versión de parche preferida actual disponible de Redis OSS 6.2 en el momento de su creación o modificación. Al especificar la versión 6.x del motor en la API se traducirá en la última versión secundaria de Redis OSS 6.

Para los clústeres 6.0 existentes, puede optar por la siguiente actualización automática de la versión secundaria configurando el parámetro `AutoMinorVersionUpgrade` a `yes` en las API `CreateCacheCluster`, `ModifyCacheCluster`, `CreateReplicationGroup` o `ModifyReplicationGroup`. ElastiCache actualizará la versión secundaria de los clústeres 6.0 existentes a 6.2 mediante actualizaciones de autoservicio. Para obtener más información, consulte Actualizaciones de autoservicio de Amazon ElastiCache.

Al llamar a la API DescribeCacheEngineVersions, el valor del parámetro `EngineVersion` se establecerá en 6.2 y la versión real del motor con la versión de parche se devolverá en el campo `CacheEngineVersionDescription`.

Para obtener más información sobre la versión 6.2 de Redis OSS, consulte Redis OSS 6.2 Release Notes en Redis OSS en GitHub.

Versión 6.0 de ElastiCache para Redis OSS (mejorada)

Amazon ElastiCache presenta la nueva versión de ElastiCache para el motor de Redis OSS, que incluye Autenticación de usuarios con control de acceso basado en roles, almacenamiento en caché en el cliente y mejoras operativas significativas.

A partir de Redis OSS 6.0, ElastiCache ofrecerá una sola versión para cada versión secundaria de Redis OSS, en lugar de ofrecer varias versiones de parches. ElastiCache también administrará automáticamente la versión de parches de los clústeres en ejecución, lo que garantiza un mejor rendimiento y más seguridad.

También puede optar por la siguiente actualización automática de la versión secundaria configurando el parámetro `AutoMinorVersionUpgrade` a `yes`. ElastiCache administrará la actualización de la versión secundaria mediante actualizaciones de autoservicio. Para obtener más información, consulte Actualizaciones de servicios en ElastiCache.

Para especificar la versión del motor, utilice `6.0`. ElastiCache invocará automáticamente la versión del parche preferida de Redis OSS 6.0 que se encuentre disponible. Por ejemplo, al crear o modificar un clúster, establece el parámetro `--engine-version` en 6.0. El clúster se lanzará con la versión de parche preferida actual disponible de Redis OSS 6.0 en el momento de su creación o modificación. Cualquier solicitud con un valor de versión de parche específico se rechazará, se lanzará una excepción y el proceso fallará.

Al llamar a la API DescribeCacheEngineVersions, el valor del parámetro `EngineVersion` se establecerá en 6.0 y la versión real del motor con la versión de parche se devolverá en el campo `CacheEngineVersionDescription`.

Para obtener más información sobre la versión 6.0 de Redis OSS, consulte Redis OSS 6.0 Release Notes en Redis OSS en GitHub.

Versión 5.0.6 de ElastiCache para Redis OSS (mejorada)

Amazon ElastiCache presenta la siguiente versión de ElastiCache para el motor de Redis OSS, que incluye correcciones de errores y las siguientes actualizaciones acumulativas:

- Garantía de estabilidad del motor en condiciones especiales.
- Control de errores HyperLogLog mejorado.
- Comandos de protocolo de enlace mejorados para reproducción de confianza.
- Seguimiento de entrega de mensajes coherente a través de comando `XCLAIM`.
- Gestión de campo de `LFU` mejorada en objetos.
- Gestión de transacciones mejorada al utilizar `ZPOP`.
- Capacidad para cambiar el nombre de comandos: un parámetro llamado `rename-commands` que le permite cambiar el nombre de los comandos de Redis OSS potencialmente peligrosos o costosos que podrían provocar una pérdida de datos accidental, por ejemplo, `FLUSHALL`

o FLUSHDB. Esto equivale a la configuración rename-command en Redis OSS de código abierto. Sin embargo, ElastiCache ha mejorado la experiencia al proporcionar un flujo de trabajo completamente administrado. Los cambios al nombre de comandos se aplican de inmediato y se propagan automáticamente en todos los nodos del clúster que contienen la lista de comandos. No es necesaria la intervención del usuario, como reiniciar nodos.

Los siguientes ejemplos muestran cómo modificar los grupos de parámetros existentes. Incluyen el parámetro `rename-commands`, que es una lista de comandos separados por espacios cuyo nombre desea cambiar:

```
aws elasticache modify-cache-parameter-group --cache-parameter-group-
name custom_param_group
--parameter-name-values "ParameterName=rename-commands,  ParameterValue='flushall
 restrictedflushall'" --region region
```

En este ejemplo, el parámetro rename-commands se utiliza para cambiar el nombre del comando `flushall` a `restrictedflushall`.

Para cambiar el nombre de varios comandos, utilice lo siguiente:

```
aws elasticache modify-cache-parameter-group --cache-parameter-group-
name custom_param_group
--parameter-name-values "ParameterName=rename-commands,  ParameterValue='flushall
 restrictedflushall flushdb restrictedflushdb'" --region region
```

Para revertir un cambio, vuelva a ejecutar el comando y excluya los valores a los que se ha cambiado el nombre de la lista `ParameterValue` que desea conservar, tal y como se muestra a continuación:

```
aws elasticache modify-cache-parameter-group --cache-parameter-group-
name custom_param_group
--parameter-name-values "ParameterName=rename-commands,  ParameterValue='flushall
 restrictedflushall'" --region region
```

En este caso, el nombre del comando `flushall` cambiará a `restrictedflushall` y los nombres de los demás comandos revertirán al nombre original.

> **Note**
>
> Al cambiar el nombre de los comandos, estará sometido a las siguientes limitaciones:
>
> - Todos los nombres de comandos cambiados deben ser alfanuméricos.
> - La longitud máxima de los nombres de comandos nuevos es de 20 caracteres alfanuméricos.
> - Al cambiar el nombre de comandos, asegúrese de actualizar el grupo de parámetros asociados con el clúster.
> - Para evitar por completo el uso de un comando, utilice la palabra clave `blocked`, tal y como se muestra a continuación:
>
> ```
> aws elasticache modify-cache-parameter-group --cache-parameter-group-
> name custom_param_group
> --parameter-name-values "ParameterName=rename-commands,
>  ParameterValue='flushall blocked'" --region region
> ```

Para obtener más información acerca de los cambios de los parámetros y obtener una lista de los comandos elegibles para el cambio de nombre, consulte Cambios de parámetros de Redis OSS 5.0.3.

- Flujos de Redis OSS: modela una estructura de datos de registro que permite a los productores añadir nuevos elementos en tiempo real. También permite a los clientes consumir mensajes mediante bloqueo o sin bloqueo. Las secuencias también permiten a los grupos de consumidores, que representan a un grupo de clientes, consumir cooperativamente distintas partes de la misma secuencia de mensajes, de forma similar a como se hace en Apache Kafka. Para obtener más información, consulte Streams.
- Compatibilidad con una familia de comandos de secuencias, como `XADD`, `XRANGE` y `XREAD`. Para obtener más información, consulte Streams Commands.
- Una serie de parámetros nuevos y con nombres nuevos. Para obtener más información, consulte Cambios de parámetros de Redis OSS 5.0.0.
- Una nueva métrica de Redis OSS, `StreamBasedCmds`.
- Tiempo de instantánea ligeramente más rápido para los nodos de Redis OSS.

> **Important**
>
> ElastiCache incorpora dos correcciones de errores críticos de la Redis OSS open source version 5.0.1. Son las siguientes:
>
> - Respuesta discrepante de RESTORE cuando algunas claves ya han vencido.
> - El comando `XCLAIM` podría devolver una entrada incorrecta o desincronizar el protocolo.
>
> Estas dos correcciones de errores se incluyen en el soporte de ElastiCache para Redis OSS para el motor de Redis versión 5.0.0 y se agregarán en actualizaciones futuras de la versión.

Para obtener más información, consulte Redis OSS 5.0.6 Release Notes en Redis OSS en GitHub.

ElastiCache versión 5.0.5 para Redis OSS (obsoleta, use la versión 5.0.6)

Amazon ElastiCache presenta la próxima versión de ElastiCache del motor de Redis OSS. Incluye cambios en la configuración en línea de ElastiCache en los clústeres de conmutación por error durante todas las operaciones planificadas. Ahora, puede escalar el clúster, actualizar la versión del motor de Redis OSS y aplicar parches y actualizaciones de mantenimiento mientras el clúster permanece en línea y continúa atendiendo las solicitudes entrantes. También incluye correcciones de errores.

Para obtener más información, consulte Redis OSS 5.0.5 Release Notes en Redis OSS en GitHub.

ElastiCache versión 5.0.4 para Redis OSS (obsoleta, use la versión 5.0.6)

Amazon ElastiCache introduce la próxima versión del motor de Redis OSS compatible con ElastiCache. Contiene las mejoras siguientes:

- Garantía de estabilidad del motor en condiciones especiales.
- Control de errores HyperLogLog mejorado.
- Comandos de protocolo de enlace mejorados para reproducción de confianza.
- Seguimiento de entrega de mensajes coherente a través de comando `XCLAIM`.
- Gestión de campo de `LFU` mejorada en objetos.
- Gestión de transacciones mejorada al utilizar `ZPOP`.

Para obtener más información, consulte Redis OSS 5.0.4 Release Notes en Redis OSS en GitHub.

ElastiCache versión 5.0.3 para Redis OSS (obsoleta, use la versión 5.0.6)

Amazon ElastiCache presenta la siguiente versión de ElastiCache para el motor de Redis OSS, en la que se han corregido errores.

ElastiCache versión 5.0.0 para Redis OSS (obsoleta, use la versión 5.0.6)

Amazon ElastiCache presenta la próxima versión principal de ElastiCache para el motor de Redis OSS. ElastiCache versión 5.0.0 para Redis OSS aporta compatibilidad con las siguientes mejoras:

- Flujos de Redis OSS: modela una estructura de datos de registro que permite a los productores añadir nuevos elementos en tiempo real. También permite a los clientes consumir mensajes mediante bloqueo o sin bloqueo. Las secuencias también permiten a los grupos de consumidores, que representan a un grupo de clientes, consumir cooperativamente distintas partes de la misma secuencia de mensajes, de forma similar a como se hace en Apache Kafka. Para obtener más información, consulte Streams.
- Compatibilidad con una familia de comandos de secuencias, como `XADD`, `XRANGE` y `XREAD`. Para obtener más información, consulte Streams Commands.
- Una serie de parámetros nuevos y con nombres nuevos. Para obtener más información, consulte Cambios de parámetros de Redis OSS 5.0.0.
- Una nueva métrica de Redis OSS, `StreamBasedCmds`.
- Tiempo de instantánea ligeramente más rápido para los nodos de Redis OSS.

Versión 4.0.10 de ElastiCache para Redis OSS (mejorada)

Amazon ElastiCache presenta la próxima versión principal de ElastiCache para el motor de Redis OSS. ElastiCache versión 4.0.10 para Redis OSS aporta compatibilidad con las siguientes mejoras:

- Cambio de tamaño y cifrado de clústeres en línea en una sola versión de ElastiCache. Para obtener más información, consulte los siguientes temas:
  - Escalado de clústeres de Valkey o Redis OSS (modo de clúster habilitado)
  - Cambio de particiones en línea para Valkey o Redis OSS (modo de clúster habilitado)
  - Seguridad de datos en Amazon ElastiCache
- Una serie de parámetros nuevos. Para obtener más información, consulte Cambios de parámetros de Redis OSS 4.0.10.

- Compatibilidad con la familia de comandos de memoria, como `MEMORY`. Para obtener más información, consulte Commands (realice una búsqueda en MEMO).
- Compatibilidad con la desfragmentación de memoria mientras está online, lo que permite un uso de memoria más eficiente y más disponibilidad de memoria para los datos.
- Compatibilidad para vaciar y eliminar de forma asincrónica. ElastiCache para Redis OSS admite comandos como `UNLINK`, `FLUSHDB` y `FLUSHALL` para que se ejecuten en un subproceso diferente del subproceso principal. Hacer esto ayuda a mejorar el rendimiento y los tiempos de respuesta para sus aplicaciones mediante la liberación de memoria de forma asincrónica.
- Una nueva métrica de Redis OSS, `ActiveDefragHits`. Para obtener más información, consulte Metrics for Redis OSS.

Los usuarios de Redis OSS (modo de clúster deshabilitado) que ejecutan ElastiCache versión 3.2.10 para Redis OSS pueden utilizar la consola para actualizar los clústeres mediante la actualización en línea.

Comparación del cambio de tamaño de clúster de ElastiCache y la compatibilidad con el cifrado

Característica	3.2.6	3.2.10	4.0.10 y versiones posteriores
Cambio de tamaño de clústeres online *	No	Sí	Sí
Cifrado en tránsito **	Sí	No	Sí
Cifrado en reposo**	Sí	No	Sí
* Adición, eliminación y reequilibrado de particiones.			
** Necesario para las aplicaciones que se ajustan al estándar FedRAMP, HIPAA y PCI DSS. Para obtener más información, consulte Validación de la conformidad para Amazon ElastiCache.			

Versiones 3.x de fin de vida útil (EOL) anteriores

Versión 3.2.10 de ElastiCache para Redis OSS (mejorada)

Amazon ElastiCache presenta la próxima versión principal de ElastiCache para el motor de Redis OSS. ElastiCache versión 3.2.10 para Redis OSS (mejorada) introduce el cambio de tamaño

de clúster en línea para añadir o eliminar particiones del clúster mientras este sigue atendiendo solicitudes de E/S entrantes. Los usuarios de ElastiCache para Redis OSS 3.2.10 tienen toda la funcionalidad de las versiones anteriores de Redis OSS, excepto la capacidad de cifrar los datos. Esta capacidad está disponible actualmente solo en la versión 3.2.6.

Comparación de las versiones 3.2.6 y 3.2.10 de ElastiCache para Redis OSS

Característica	3.2.6	3.2.10
Cambio de tamaño de clústeres online *	No	Sí
Cifrado en tránsito **	Sí	No
Cifrado en reposo**	Sí	No
* Adición, eliminación y reequilibrado de particiones.		
** Necesario para las aplicaciones que se ajustan al estándar FedRAMP, HIPAA y PCI DSS. Para obtener más información, consulte Validación de la conformidad para Amazon ElastiCache.		

Para obtener más información, consulte los siguientes temas:

- Cambio de particiones en línea para Valkey o Redis OSS (modo de clúster habilitado)
- Cambio de tamaño de clústeres online

Versión 3.2.6 de ElastiCache para Redis OSS (mejorada)

Amazon ElastiCache presenta la próxima versión principal de ElastiCache para el motor de Redis OSS. Los usuarios de la versión 3.2.6 d ElastiCache para Redis OSS tienen toda la funcionalidad de las versiones anteriores de Redis OSS más la opción de cifrar los datos. Para obtener más información, consulte los siguientes temas:

- Cifrado en tránsito (TLS) de ElastiCache
- Cifrado en reposo en ElastiCache
- Validación de la conformidad para Amazon ElastiCache

Versión 3.2.4 de ElastiCache para Redis OSS (mejorada)

Amazon ElastiCache versión 3.2.4 presenta la próxima versión principal de ElastiCache para el motor de Redis OSS. Los usuarios de ElastiCache 3.2.4 cuentan con todas las funcionalidades de las versiones anteriores de Redis OSS junto con la posibilidad de utilizar el modo de clúster o en el modo sin clúster. La tabla siguiente ofrece un resumen.

Comparación del modo de clúster y sin clúster de Redis OSS 3.2.4

Característica	Modo sin clúster	Modo clúster
Particiones de datos	No	Sí
Indexación geoespacial	Sí	Sí
Cambio de tipo de nodo	Sí	Sí *
Escalado de réplica	Sí	Sí *
Escalado ascendente	No	Sí *
Compatibilidad con bases de datos	Múltiple	Única
Grupo de parámetros	`default.redis3.2` **	`default.redis3.2.cluster.on` **

* Consulte Restauración desde una copia de seguridad a una nueva caché

** O uno derivado.

Notas:

- Particionamiento: posibilidad de dividir los datos entre 2 a 500 grupos de nodos (particiones) con compatibilidad de reproducción para cada grupo de nodos.
- Indexación geoespacial: Redis OSS 3.2.4 introduce la compatibilidad con la indexación geoespacial a través de seis comandos de GEO. Para obtener más información, consulte la documentación de comandos de GEO* de Redis OSS Commands: GEO en la página de comandos de Valkey (filtrado para GEO).

Para obtener más información acerca de las características adicionales de Redis OSS 3, consulte Redis OSS 3.2 release notes y Redis OSS 3.0 release notes.

Actualmente, Valkey o Redis OSS (modo de clúster habilitado) administrados por ElastiCache no son compatibles con las siguientes características de Redis OSS 3.2:

- Migración de réplica
- Reequilibrado de clúster
- Depurador Lua

ElastiCache deshabilita los siguientes comandos de administración de Redis OSS 3.2:

- `cluster meet`
- `cluster replicate`
- `cluster flushslots`
- `cluster addslots`
- `cluster delslots`
- `cluster setslot`
- `cluster saveconfig`
- `cluster forget`
- `cluster failover`
- `cluster bumpepoch`
- `cluster set-config-epoch`
- `cluster reset`

Para obtener más información acerca de los parámetros de Redis OSS 3.2.4, consulte Cambios de parámetros de Redis OSS 3.2.4.

Versiones 2.x de fin de vida útil (EOL) anteriores

Versión 2.8.24 de ElastiCache para Redis OSS (mejorada)

Entre las mejoras de Redis OSS añadidas desde la versión 2.8.23 se incluyen varias correcciones de errores y el registro de direcciones de acceso a la memoria incorrectas. Para obtener más información, consulte Redis OSS 2.8 release notes.

Versión 2.8.23 de ElastiCache para Redis OSS (mejorada)

Entre las mejoras de Redis OSS que se han añadido desde la versión 2.8.22, se incluyen varias correcciones de errores. Para obtener más información, consulte Redis OSS 2.8 release notes. Esta versión también incluye compatibilidad con el nuevo parámetro `close-on-slave-write` que, si está habilitado, desconecta a los clientes que intentan escribir en una réplica de solo lectura.

Para obtener más información sobre los parámetros de Redis OSS 2.8.23, consulte Parámetros que se han añadido a Redis OSS 2.8.23 (mejorado) en la Guía del usuario de ElastiCache.

Versión 2.8.22 de ElastiCache para Redis OSS (mejorada)

Entre las mejoras de Redis OSS incorporadas desde la versión 2.8.21 se incluyen las siguientes:

- Compatibilidad con las copias de seguridad y las sincronizaciones sin ramificaciones que permiten asignar menos memoria para las copias de seguridad y más para su aplicación. Para obtener más información, consulte Cómo se implementan la sincronización y la copia de seguridad. El proceso sin ramificaciones puede afectar a la latencia y al rendimiento. En caso de elevado rendimiento de escritura, al volver a sincronizar una réplica es posible que esta no esté disponible durante la sincronización.
- Si se produce una conmutación por error, los grupos de reproducción se recuperarán ahora más rápido porque las réplicas realizarán sincronizaciones parciales con el nodo principal en lugar de sincronizaciones completas siempre que sea posible. Además, tanto el nodo principal como las réplicas ya no usan el disco durante las sincronizaciones, lo que proporciona mayor velocidad.
- Compatibilidad con dos métricas de CloudWatch nuevas.
  - `ReplicationBytes`: número de bytes que el clúster principal de un grupo de reproducción envía a las réplicas de lectura.
  - `SaveInProgress`: valor binario que indica si hay un proceso de almacenamiento en ejecución en segundo plano.

  Para obtener más información, consulte Monitoreo del uso con métricas de CloudWatch.
- Varias correcciones de errores críticos en el comportamiento PSYNC de reproducción. Para obtener más información, consulte Redis OSS 2.8 release notes.
- Para mantener un rendimiento de reproducción mejorado en los grupos de reproducción de Multi-AZ y a fin de incrementar la estabilidad del clúster, las réplicas que no son de ElastiCache han dejado de ser compatibles.
- Para mejorar la coherencia de los datos entre el clúster principal y las réplicas de un grupo de reproducción, las réplicas ya no desalojan claves independientes del clúster principal.

- Las variables de configuración de Redis OSS `appendonly` y `appendfsync` no se admiten en la versión 2.8.22 de Redis OSS y las versiones posteriores.
- En situaciones de baja memoria, es posible que los clientes que tengan un búfer de salida de gran tamaño se desconecten del clúster de réplica. En caso de desconexión, el cliente debe volver a conectarse. Estas situaciones son más probables para clientes PUBSUB.

Versión 2.8.21 de ElastiCache para Redis OSS

Entre las mejoras de Redis OSS que se han añadido desde la versión 2.8.19 se incluyen varias correcciones de errores. Para obtener más información, consulte Redis OSS 2.8 release notes.

Versión 2.8.19 de ElastiCache para Redis OSS

Entre las mejoras de Redis OSS incorporadas desde la versión 2.8.6 se incluyen las siguientes:

- Compatibilidad con HyperLogLog. Para obtener más información, consulte Redis OSS new data structure: HyperLogLog.
- El tipo de datos de conjuntos clasificados ahora es compatible con las consultas de gama lexicográfica gracias a los nuevos comandos `ZRANGEBYLEX`, `ZLEXCOUNT` y `ZREMRANGEBYLEX`.
- Para evitar que un nodo principal envíe obsoletos a los nodos de réplica, se producirá un error en la sincronización principal si se interrumpe un proceso secundario de almacenamiento en segundo plano (`bgsave`).
- Compatibilidad con la métrica de CloudWatch HyperLogLogBasedCommands. Para obtener más información, consulte Métricas de Valkey y Redis OSS.

Versión 2.8.6 de ElastiCache para Redis OSS

Entre las mejoras de Redis OSS incorporadas desde la versión 2.6.13 se incluyen las siguientes:

- Mejora de la flexibilidad y la tolerancia a errores de las réplicas de lectura.
- Compatibilidad con la resincronización.
- Compatibilidad con un número mínimo definido por el usuario de réplicas de lectura que deben estar disponibles en todo momento.
- Compatibilidad completa con los clientes de notificación de publicaciones/suscripciones de eventos en el servidor.
- Detección automática de errores en un nodo principal y de conmutaciones por error del nodo principal al nodo secundario.

Versión 2.6.13 de ElastiCache para Redis OSS

La versión 2.6.13 de ElastiCache para Redis OSS fue la primera versión de ElastiCache compatible con Redis OSS. Multi-AZ no es compatible con la versión 2.6.13 de ElastiCache para Redis OSS.

Programa de fin de vida útil de las versiones de ElastiCache para Redis OSS

En esta sección se definen las fechas de fin de vida útil (EOL) de las versiones principales más antiguas a medida que se anuncian. Esto le permite tomar decisiones sobre versiones y actualizaciones para el futuro.

> **Note**
>
> Las versiones de ElastiCache de 5.0.0 a 5.0.5 para Redis OSS están obsoletas. Utilice las versiones 5.0.6 o superior.

En la siguiente tabla se muestra el programa de Soporte extendido para los motores de ElastiCache para Redis OSS.

Programación del soporte extendido y del fin de la vida útil

Versión principal del motor	Fecha de finalización del soporte estándar	Inicio del soporte extendido Y1 Premium	Inicio del soporte extendido Y2 Premium	Inicio del soporte extendido Y3 Premium	Fin del soporte extendido y versión EOL
Redis OSS versión 4	31/1/2026	1/2/2026	1/2/2027	1/2/2028	31/1/2029
Redis OSS versión 5	31/1/2026	1/2/2026	1/2/2027	1/2/2028	31/1/2029
Redis OSS versión 6	31/1/2027	1/2/2027	1/2/2028	1/2/2029	31/1/2030

La siguiente tabla resume cada versión y la fecha de fin de vida anunciada, así como la versión de destino de actualización recomendada.

EOL anterior

V p d o	Versiones secundarias de origen	Destino de actualización recomendado	Fecha de EOL
V	3.2.4, 3.2.6 y 3.2.10	Versión 6.2 o superior **Note** Para las regiones US-ISO-EAST-1, US-ISO-WEST-1 y US-ISOB-EAST-1, recomendamos 5.0.6 o superior.	31 de julio de 2023
V	2.8.24, 2.8.23, 2.8.22, 2.8.21, 2.8.19, 2.8.12, 2.8.6, 2.6.13	Versión 6.2 o superior **Note** Para las regiones US-ISO-EAST-1, US-ISO-WEST-1 y US-ISOB-EAST-1, recomendamos 5.0.6 o superior.	13 de enero de 2023

Versiones compatibles de ElastiCache para Memcached

ElastiCache es compatible con las siguientes versiones de Memcached y permite actualizar a versiones más recientes. Cuando actualice a una versión más reciente, preste especial atención a las condiciones que, si no se cumplen, harán que se produzca un error de actualización.

Versiones de ElastiCache para Memcached

- ElastiCache versión 1.6.22 para Memcached
- ElastiCache versión 1.6.17 para Memcached
- ElastiCache versión 1.6.12 para Memcached
- ElastiCache versión 1.6.6 para Memcached
- ElastiCache versión 1.5.16 para Memcached
- ElastiCache versión 1.5.10 para Memcached
- ElastiCache versión 1.4.34 para Memcached
- ElastiCache versión 1.4.33 para Memcached
- ElastiCache versión 1.4.24 para Memcached
- ElastiCache versión 1.4.14 para Memcached
- ElastiCache versión 1.4.5 para Memcached

ElastiCache versión 1.6.22 para Memcached

ElastiCache para Memcached versión 1.6.22 agrega soporte para la versión 1.6.22 de Memcached. No incluye características nuevas, pero sí correcciones de errores y actualizaciones acumulativas de Memcached 1.6.18.

Para obtener más información, consulte ReleaseNotes1622 en Memcached en GitHub.

ElastiCache versión 1.6.17 para Memcached

ElastiCache para Memcached versión 1.6.17 agrega soporte para la versión 1.6.17 de Memcached. No incluye características nuevas, pero sí correcciones de errores y actualizaciones acumulativas de Memcached 1.6.17.

Para obtener más información, consulte ReleaseNotes1617 en Memcached en GitHub.

ElastiCache versión 1.6.12 para Memcached

ElastiCache para Memcached versión 1.6.12 agrega soporte para la versión 1.6.12 de Memcached y el cifrado en tránsito. Incluye también correcciones de errores y actualizaciones acumulativas desde Memcached 1.6.6.

Para obtener más información, consulte Notas de la versión 1612 en Memcached en GitHub.

ElastiCache versión 1.6.6 para Memcached

ElastiCache para Memcached versión 1.6.6 agrega soporte para la versión 1.6.6 de Memcached. No incluye características nuevas, pero sí correcciones de errores y actualizaciones acumulativas de Memcached 1.5.16. ElastiCache para Memcached no incluye compatibilidad con Extstore.

Para obtener más información, consulte Notas de la versión 166 en Memcached en GitHub.

ElastiCache versión 1.5.16 para Memcached

ElastiCache versión 1.5.16 para Memcached agrega soporte para la versión 1.5.16 de Memcached. No incluye características, pero sí correcciones de errores y actualizaciones acumulativas de Memcached 1.5.14 y Memcached 1.5.15.

Para obtener más información, consulte Notas de la versión de Memcached 1.5.16 en Memcached en GitHub.

ElastiCache versión 1.5.10 para Memcached

ElastiCache versión 1.5.10 para Memcached admite las siguientes características de Memcached:

- Reequilibrado automático de slabs.
- Búsquedas más rápidas en tablas hash con el algoritmo `murmur3`.
- Algoritmo LRU segmentado.
- LRU crawler para realizar la operación background-reclaim con la memoria.
- `--enable-seccomp`: opción de tiempo de compilación.

También introduce los parámetros `no_modern` e `inline_ascii_resp`. Para obtener más información, consulte Cambios en los parámetros de Memcached 1.5.10.

Entre las mejoras de Memcached incorporadas desde la versión 1.4.34 de ElastiCache para Memcached se incluyen las siguientes:

- Correcciones acumulativas, tales como multigets ASCII, rastreos CVE-2017-9951 y de límite para `metadumper`.
- Mejor administración de conexiones al cerrar las conexiones que alcanzan el límite de conexión.
- Se ha mejorado la administración del tamaño de los elementos para tamaños de elementos superiores a 1 MB.
- Mejoras en el rendimiento y en la sobrecarga de memoria gracias a la reducción de los requisitos de memoria por elemento en unos pocos bytes.

Para obtener más información, consulte Notas de la versión de Memcached 1.5.10 en Memcached en GitHub.

ElastiCache versión 1.4.34 para Memcached

ElastiCache versión 1.4.34 para Memcached no agrega características nuevas a la versión 1.4.33. La versión 1.4.34 es una versión de corrección de errores que es mayor que las versiones habituales de este tipo.

Para obtener más información, consulte Notas de la versión de Memcached 1.4.34 en Memcached en GitHub.

ElastiCache versión 1.4.33 para Memcached

Entre las mejoras incorporadas desde la versión 1.4.24 se incluyen las siguientes:

- Capacidad para volcar todos los metadatos para una determinada clase slab, una lista de clases slab o todas las clases slab. Para obtener más información, consulte Memcached 1.4.31 Release Notes.
- Compatibilidad mejorada con elementos de gran tamaño superiores al tamaño predeterminado de 1 megabyte. Para obtener más información, consulte Memcached 1.4.29 Release Notes.
- Posibilidad de especificar el tiempo durante el cual un cliente puede permanecer inactivo antes de solicitar su cierre.

  Posibilidad para incrementar de forma dinámica la cantidad de memoria disponible para Memcached sin tener que reiniciar el clúster. Para obtener más información, consulte Memcached 1.4.27 Release Notes.
- Compatibilidad con el registro de `fetchers`, `mutations` y `evictions`. Para obtener más información, consulte Memcached 1.4.26 Release Notes.

- Posibilidad de reclamar la memoria liberada en un grupo global para reasignarla a nuevas clases slab. Para obtener más información, consulte Memcached 1.4.25 Release Notes.
- Varias correcciones de errores.
- Incorporación de algunos comandos y parámetros nuevos. Para ver una lista, consulte Parámetros agregados a Memcached 1.4.33.

ElastiCache versión 1.4.24 para Memcached

Entre las mejoras incorporadas desde la versión 1.4.14 se incluyen las siguientes:

- Administración de elementos menos usados recientemente (LRU) mediante un proceso en segundo plano.
- Se ha añadido la posibilidad de usar jenkins o murmur3 como algoritmo hash.
- Incorporación de algunos comandos y parámetros nuevos. Para ver una lista, consulte Parámetros agregados a Memcached 1.4.24.
- Varias correcciones de errores.

ElastiCache versión 1.4.14 para Memcached

Entre las mejoras incorporadas desde la versión 1.4.5 se incluyen las siguientes:

- Capacidad mejorada para reequilibrar slab.
- Mejora del rendimiento y la escalabilidad.
- Se ha introducido el comando touch que permite actualizar hora de vencimiento de un elemento existente sin necesidad de recuperarlo.
- Detección automática: capacidad para los programas de cliente de determinar automáticamente todos los nodos de caché en un clúster e iniciar y mantener las conexiones a todos estos nodos.

ElastiCache versión 1.4.5 para Memcached

ElastiCache versión 1.4.5 para Memcached fue el primer motor y la primera versión compatible con Amazon ElastiCache para Memcached.

Diferencias de compatibilidad y comportamiento de versiones principales del motor con Valkey

Valkey 7.2.6 presenta diferencias de compatibilidad similares con las versiones anteriores de Redis OSS 7.2.4. Para ver la versión compatible más reciente de Valkey, consulte Motores y versiones compatibles.

Para obtener más información sobre la versión 7.2 de Valkey, consulte Redis OSS 7.2.4 Release Notes (Valkey 7.2 incluye todos los cambios de Redis OSS hasta la versión 7.2.4) y Valkey 7.2 release notes en Valkey en GitHub.

Estos son los posibles cambios de comportamiento importantes entre Valkey 7.2 y Redis OSS 7.1 (o 7.0):

- El muestreo temporal se bloquea durante la ejecución de comandos y en los scripts.
- Un comando de flujo bloqueado que se ejecuta cuando la clave ya no existe contiene un código de error diferente (-NOGROUP o -WRONGTYPE en lugar de -UNBLOCKED).
- El seguimiento de scripts por parte del cliente ahora rastrea las claves que lee el script, en lugar de las claves declaradas por la persona que llama a EVAL/FCALL.

Diferencias de compatibilidad y comportamiento de las versiones principales del motor con Redis OSS

> **Important**
>
> La siguiente página está estructurada para indicar todas las diferencias de incompatibilidad entre las versiones e informarle de las consideraciones que debe tener en cuenta al actualizar a versiones más recientes. Esta lista incluye cualquier problema de incompatibilidad de versiones que pueda encontrar al actualizar.
> Puede actualizar directamente desde la versión actual de Redis OSS a la última versión de Redis OSS disponible, sin necesidad de realizar actualizaciones secuenciales. Por ejemplo, puede actualizar directamente desde la versión 3.0 de Redis OSS a la versión 7.0.

Las versiones de Redis OSS se identifican con una versión semántica que consta de un componente principal, secundario y de parches. Por ejemplo, en Redis OSS 4.0.10, la versión principal es 4, la

versión secundaria 0 y la versión del parche es 10. Por lo general, estos valores se incrementan en función de las siguientes convenciones:

- Las versiones principales son para cambios incompatibles con la API.
- Las versiones secundarias son para nuevas funciones añadidas de manera compatible con versiones anteriores.
- Las versiones de parche son para correcciones de errores compatibles con versiones anteriores y cambios no funcionales.

Recomendamos mantener siempre la última versión del parche dentro de una versión major.minor determinada para tener las mejoras de rendimiento y estabilidad más recientes. A partir de ElastiCache versión 6.0 para Redis OSS, ElastiCache ofrecerá una sola versión para cada versión secundaria de Redis OSS, en lugar de ofrecer varias versiones de parches. ElastiCache también administrará automáticamente la versión de parches de los clústeres en ejecución, lo que garantiza un mejor rendimiento y más seguridad.

También recomendamos actualizar periódicamente a la última versión principal, ya que la mayoría de las mejoras importantes no se transfieren a versiones anteriores. Cuando ElastiCache amplía su disponibilidad a una nueva región de AWS, ElastiCache para Redis admite las dos versiones major.minor más recientes en ese momento para la nueva región. Por ejemplo, si se lanza una nueva región de AWS y las versiones major.minor más recientes de ElastiCache para Redis son 7.0 y 6.2, ElastiCache admitirá las versiones 7.0 y 6.2 de Redis OSS en la nueva región de AWS. A medida que se publiquen nuevas versiones major.minor de ElastiCache para Redis OSS, ElastiCache continuará añadiendo compatibilidad con las versiones más recientes. Para obtener más información sobre cómo elegir regiones para ElastiCache, consulte Elección de regiones y zonas de disponibilidad.

Cuando realice una actualización que abarque versiones principales o secundarias, tenga en cuenta la siguiente lista, que incluye el comportamiento y los cambios incompatibles con versiones anteriores publicados con Redis OSS a lo largo del tiempo.

Cambios de comportamiento e incompatibles con versiones anteriores de Redis OSS 7.0

Para obtener una lista completa de los cambios, consulte Redis OSS 7.0 release notes.

- `SCRIPT LOAD` y `SCRIPT FLUSH` ya no se propagan a réplicas. Si necesita algo de durabilidad para los scripts, le recomendamos que considere utilizar funciones de Redis OSS.

- Los canales de pubsub ahora están bloqueados de forma predeterminada para los nuevos usuarios de ACL.
- El comando `STRALGO` se sustituyó por el comando `LCS`.
- El formato de `ACL GETUSER` ha cambiado para que todos los campos muestren el patrón de cadena de acceso estándar. Si utilizaba la automatización mediante `ACL GETUSER`, debe comprobar que maneje cualquiera de los formatos.
- Las categorías de ACL para `SELECT`, `WAIT`, `ROLE`, `LASTSAVE`, `READONLY`, `READWRITE` y `ASKING` han cambiado.
- El comando `INFO` ahora muestra las estadísticas de los comandos por subcomando en lugar de en los comandos del contenedor de nivel superior.
- Los valores devueltos de los comandos `LPOP`, `RPOP`, `ZPOPMIN` y `ZPOPMAX` han cambiado en determinados casos límite. Si utiliza estos comandos, debe consultar las notas de la versión y evaluar si se ve afectado.
- Los comandos `SORT` y `SORT_RO` ahora requieren acceso a todo el espacio de claves para poder utilizar los argumentos `GET` y `BY`.

Cambios de comportamiento e incompatibles con versiones anteriores de Redis OSS 6.2

Para obtener una lista completa de los cambios, consulte Redis OSS 6.2 release notes.

- Los indicadores de ACL de los comandos `TIME`, `ECHO`, `ROLE` y `LASTSAVE` se cambiaron. Esto puede provocar que se rechacen los comandos previamente permitidos y viceversa.

> **Note**
>
> Ninguno de estos comandos modifica ni da acceso a los datos.

- Al actualizar desde Redis OSS 6.0, se cambia el orden de los pares clave/valor devueltos por una respuesta de mapa a un script de lua. Si sus scripts utilizan `redis.setresp()` o devuelven un mapa (nueva funcionalidad en Redis OSS 6.0), tenga en cuenta las implicaciones de que el script deje de funcionar en las actualizaciones.

Cambios de comportamiento e incompatibles con versiones anteriores de Redis OSS 6.0

Para obtener una lista completa de los cambios, consulte Redis OSS 6.0 release notes.

- El número máximo de bases de datos permitidas se ha reducido de 1,2 millones a 10 mil. El valor predeterminado es 16 y no recomendamos el uso de valores mucho mayores, ya que hemos detectado problemas de rendimiento y memoria.
- Establezca el parámetro `AutoMinorVersionUpgrade` en “sí”. ElastiCache administrará la actualización de la versión secundaria mediante actualizaciones de autoservicio. Esto se gestionará a través de canales de notificación estándar a los clientes mediante una campaña de actualización de autoservicio. Para obtener más información, consulte Actualizaciones de autoservicio de ElastiCache.

Cambios de comportamiento e incompatibles con versiones anteriores de Redis OSS 5.0

Para obtener una lista completa de los cambios, consulte Redis OSS 5.0 release notes.

- Los scripts se replican por efectos en lugar de volver a ejecutar el script en la réplica. Por lo general, esto mejora el rendimiento, pero puede aumentar la cantidad de datos replicados entre los principales y las réplicas. Hay una opción para volver al comportamiento anterior que solo está disponible en ElastiCache versión 5.0 para Redis OSS.
- Si está actualizando desde Redis OSS 4.0, algunos comandos de los scripts de LUA devolverán los argumentos en un orden diferente al de las versiones anteriores. En Redis OSS 4.0, Redis ordenaba algunas respuestas de forma lexográfica para hacer que fueran deterministas; este orden no se aplica cuando los scripts se replican por efectos.
- En Redis OSS 5.0.3 y versiones posteriores, ElastiCache para Redis OSS descargará parte del trabajo de E/S a los núcleos en segundo plano en los tipos de instancia con más de 4 VCPU. Esto puede cambiar las características de rendimiento de Redis OSS y los valores de algunas métricas. Para obtener más información, consulte ¿Qué métricas debo monitorear? para saber si necesita cambiar las métricas que ve.

Cambios de comportamiento e incompatibles con versiones anteriores de Redis OSS 4.0

Para obtener una lista completa de los cambios, consulte Redis OSS 4.0 release notes.

- El registro lento ahora registra dos argumentos adicionales, el nombre y la dirección del cliente. Este cambio debe ser compatible con versiones anteriores a menos que dependa explícitamente de cada entrada de registro lento que contenga 3 valores.
- El comando `CLUSTER NODES` ahora devuelve un formato ligeramente diferente, que no es compatible con versiones anteriores. Recomendamos que los clientes no usen este comando para obtener información sobre los nodos presentes en un clúster y que, en su lugar, usen `CLUSTER SLOTS`.

EOL anterior

Cambios de comportamiento e incompatibles con versiones anteriores de Redis OSS 3.2

Para obtener una lista completa de los cambios, consulte Redis OSS 3.2 release notes.

- No hay cambios de compatibilidad que destacar para esta versión.

Para obtener más información, consulte Programa de fin de vida útil de las versiones de ElastiCache para Redis OSS.

Cambios de comportamiento e incompatibles con versiones anteriores de Redis OSS 2.8

Para obtener una lista completa de los cambios, consulte Redis OSS 2.8 release notes.

- A partir de Redis OSS 2.8.22, Redis OSS AOF ya no es compatible con ElastiCache para Redis OSS. Recomendamos usar MemoryDB cuando sea necesario conservar los datos de forma duradera.
- A partir de Redis OSS 2.8.22, ElastiCache para Redis OSS ya no admite la conexión de réplicas a los componentes principales alojados en ElastiCache. Durante la actualización, las réplicas externas se desconectarán y no podrán volver a conectarse. Recomendamos utilizar el almacenamiento en caché en el cliente, disponible en Redis OSS 6.0, como alternativa a las réplicas externas.

- Los comandos TTL y PTTL ahora devuelven -2 si la clave no existe y -1 si existe pero no tiene fecha de caducidad asociada. En Redis OSS 2.6 y versiones anteriores, se devolvía -1 para ambas condiciones.
- SORT con ALPHA ahora ordena según la configuración regional de intercalación local si no se utiliza la opción STORE.

Para obtener más información, consulte Programa de fin de vida útil de las versiones de ElastiCache para Redis OSS.

Consideraciones sobre la actualización al trabajar con clústeres basados en nodos

> **Note**
>
> Las siguientes consideraciones solo son aplicables al actualizar clústeres basados en nodos. No son aplicables a ElastiCache Serverless.

Consideraciones sobre Valkey y Redis OSS

Cuando actualice un clúster basado en nodo de Valkey o Redis OSS, tenga en cuenta lo siguiente.

- La administración de la versión del motor está diseñada para que pueda tener el mayor control posible sobre cómo se produce la aplicación de parches. Sin embargo, ElastiCache se reserva el derecho de aplicar parches en el clúster en su nombre, llegado el caso improbable de una vulnerabilidad de seguridad crítica en el sistema o el software de caché.
- A partir de ElastiCache versión 7.2 para Valkey y ElastiCache versión 6.0 para Redis OSS, ElastiCache ofrecerá una sola versión para cada versión secundaria, en lugar de ofrecer varias versiones de parche.
- A partir de la versión 5.0.6 del motor de Redis OSS, puede actualizar la versión del clúster con un tiempo de inactividad mínimo. El clúster está disponible para operaciones de lectura durante toda la actualización y para operaciones de escritura durante la mayoría del proceso, excepto durante la operación de conmutación por error, que dura unos segundos.
- También puede actualizar los clústeres de ElastiCache con versiones anteriores a la 5.0.6. El proceso involucrado es el mismo, pero puede incurrir en un tiempo de conmutación por error más largo durante la propagación de DNS (de 30 s a 1 m).

- A partir de Redis OSS 7, ElastiCache admite el cambio entre Valkey o Redis OSS (modo de clúster deshabilitado) y Valkey o Redis OSS (modo de clúster habilitado).
- El proceso de actualización del motor de Amazon ElastiCache para Redis OSS se ha diseñado para hacer todo lo posible por retener los datos existentes y requiere que la replicación de Redis OSS se realice correctamente.
- Al actualizar el motor, ElastiCache cancelará las conexiones de cliente existentes. Para minimizar el tiempo de inactividad durante las actualizaciones del motor, le recomendamos que implemente las prácticas recomendadas para los clientes de Redis OSS, con reintentos de errores y retrocesos exponenciales, así como las prácticas recomendadas para minimizar el tiempo de inactividad durante el mantenimiento.
- No puede actualizar directamente de Valkey o Redis OSS (modo de clúster deshabilitado) a Valkey o Redis OSS (modo de clúster habilitado) cuando actualiza su motor. El siguiente procedimiento muestra cómo actualizar de Valkey o Redis OSS (modo de clúster deshabilitado) a Valkey o Redis OSS (modo de clúster habilitado).

  Actualización de una versión de motor de Valkey o Redis OSS (modo de clúster deshabilitado) a la versión del motor de Valkey o Redis OSS (modo de clúster habilitado)

  1. Realice una copia de seguridad de su clúster o grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado). Para obtener más información, consulte Copias de seguridad manuales.
  2. Utilice la copia de seguridad para crear y propagar un clúster de Valkey o Redis OSS (modo de clúster habilitado) con una partición (grupo de nodo). Especifique la nueva versión de motor y habilite el modo de clúster al crear el clúster o grupo de reproducción. Para obtener más información, consulte Tutorial: propagación de un nuevo clúster basado en nodos con una copia de seguridad creada externamente.
  3. Elimine el clúster o el grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) anterior. Para obtener más información, consulte Eliminación de un clúster en ElastiCache o Eliminación de un grupo de reproducción.
  4. Escale el nuevo grupo de replicación o clúster de Valkey o Redis OSS (modo de clúster habilitado) al número de particiones (grupos de nodo) que necesita. Para obtener más información, consulte Escalado de clústeres de Valkey o Redis OSS (modo de clúster habilitado)
- Cuando actualiza las versiones principales del motor, por ejemplo de 5.0.6 a 6.0, debe seleccionar un grupo de parámetros nuevo que sea compatible con la versión del motor nueva.

- Para clústeres de Redis OSS sencillos y clústeres con multi-AZ deshabilitado, recomendamos disponer de suficiente memoria para Redis OSS, tal y como se describe en Forma de garantizar que dispone de memoria suficiente para crear una instantánea de Valkey o Redis OSS. En estos casos, el nodo principal no está disponible para las solicitudes de servicio durante el proceso de actualización.
- Para clústeres de Redis OSS con multi-AZ habilitado, también recomendamos que programe actualizaciones del motor durante los periodos de poco tráfico entrante. Cuando se actualiza a Redis OSS 5.0.6 o a una versión posterior, el clúster principal sigue disponible para atender solicitudes durante el proceso de actualización.

  Los clústeres y grupos de reproducción con varias particiones se procesan y se aplican parches de la siguiente manera:

  - Todas las particiones se procesan en paralelo. Solo se realiza una operación de actualización en una partición a la vez.
  - En cada partición, todas las réplicas se procesan antes que el principal. Si hay menos réplicas en una partición, el principal de esa partición podrá procesarse antes que las réplicas de otras particiones terminen de procesarse.
  - En todas las particiones, los nodos principales se procesan en series. Solo se actualiza un nodo principal a la vez.
- Si el cifrado se encuentra habilitado en su grupo de reproducción o clúster actual, no puede actualizar a una versión del motor que no admita cifrado, como de la versión 3.2.6 a la 3.2.10.

Consideraciones sobre Memcached

Cuando actualice un clúster de Memcached basado en nodos, tenga en cuenta lo siguiente.

- La administración de la versión del motor está diseñada para que pueda tener el mayor control posible sobre cómo se produce la aplicación de parches. Sin embargo, ElastiCache se reserva el derecho de aplicar parches en el clúster en su nombre, llegado el caso improbable de una vulnerabilidad de seguridad crítica en el sistema o el software de caché.
- Puesto que el motor de Memcached no es compatible con la persistencia, las actualizaciones de versión del motor de Memcached son siempre un proceso disruptivo que borra todos los datos de caché del clúster.

Procedimientos recomendados y estrategias de almacenamiento en caché de ElastiCache

A continuación, puede encontrar los procedimientos recomendados para Amazon ElastiCache. Si observa estos procedimientos, mejorará el rendimiento y la fiabilidad de su caché.

Temas

- Prácticas recomendadas generales
- Prácticas recomendadas para utilizar réplicas de lectura
- Comandos de Valkey, Memcached y Redis OSS compatibles y restringidos
- Configuración y límites de Valkey y Redis OSS
- Ejemplos de clientes de IPv6 para Valkey, Memcached y Redis OSS
- Prácticas recomendadas para clientes (Valkey y Redis OSS)
- Prácticas recomendadas para clientes (Memcached)
- Clústeres de ElastiCache de doble pila compatibles con TLS
- Administración de la memoria reservada para Valkey y Redis OSS
- Prácticas recomendadas al trabajar con clústeres basados en nodos de Valkey y Redis OSS
- Estrategias de almacenamiento en caché para Memcached

Prácticas recomendadas generales

A continuación, encontrará información sobre las prácticas recomendadas para usar las interfaces de Valkey, Memcached y Redis OSS en ElastiCache.

- Utilice configuraciones habilitadas para el modo de clúster: las configuraciones habilitadas para el modo de clúster permiten que la caché se escale horizontalmente para lograr un mayor almacenamiento y rendimiento que una configuración en modo de clúster deshabilitado. ElastiCache Serverless solo está disponible en una configuración habilitada para el modo de clúster.
- Utilice conexiones de larga duración: crear una nueva conexión es caro, y requiere tiempo y recursos de CPU de la memoria caché. Reutilice las conexiones siempre que sea posible (por ejemplo, mediante la agrupación de conexiones) para amortizar este coste a lo largo de muchos comandos.

- Lea desde réplicas: si utiliza ElastiCache sin servidor o ha aprovisionado réplicas de lectura (clústeres basados en nodos), dirija las lecturas a las réplicas para lograr una mejor escalabilidad o una latencia más baja. Las lecturas desde réplicas acaban siendo coherehentes con la principal.

  En un clúster basado en nodos, evite dirigir las solicitudes de lectura a una única réplica de lectura, ya que es posible que las lecturas no estén disponibles temporalmente si el nodo falla. Configure su cliente para que dirija las solicitudes de lectura al menos a dos réplicas de lectura, o dirija las lecturas a una única réplica y a la principal.

  En ElastiCache Serverless, la lectura desde el puerto de réplica (6380) dirigirá las lecturas a la zona de disponibilidad local del cliente siempre que sea posible, lo que reducirá la latencia de recuperación. Cuando se produzca un error, volverá automáticamente a los demás nodos.
- Evitar los comandos costosos: evite ejecutar operaciones que hagan una utilización intensiva de procesamiento y de E/S, como los comandos `KEYS` y `SMEMBERS`. Recomendamos este enfoque porque estas operaciones aumentan la carga en el clúster e influyen en el rendimiento del clúster. En su lugar, utilice los comandos `SCAN` y `SSCAN`.
- Seguir las prácticas recomendadas de Lua: evite los scripts Lua de ejecución prolongada y siempre declare por adelantado las claves que utiliza en los scripts Lua. Recomendamos este enfoque para determinar que el script Lua no está utilizando comandos de ranura cruzada. Asegúrese de que las claves utilizadas en scripts Lua pertenezcan a la misma ranura.
- Utilice la publicación/suscripción en particiones: cuando utilice Valkey o Redis OSS para admitir cargas de trabajo de publicación/suscripción con un alto rendimiento, le recomendamos que utilice la publicación/suscripción en particiones (disponible con Redis OSS 7 o versiones posteriores). El pub/sub tradicional en los clústeres habilitados para el modo de clúster transmite mensajes a todos los nodos del clúster, lo que puede generar un nivel alto de `EngineCPUUtilization`. Tenga en cuenta que, en ElastiCache Serverless, los comandos pub/sub tradicionales utilizan internamente comandos pub/sub en particiones.

Prácticas recomendadas para utilizar réplicas de lectura

Muchas aplicaciones, como los almacenes de sesiones, las tablas de clasificación y los motores de recomendaciones, requieren una alta disponibilidad y gestionan muchas más operaciones de lectura que de escritura. A menudo, estas aplicaciones toleran datos ligeramente obsoletos (coherencia final), lo que significa que no ocurre nada si distintos usuarios ven momentáneamente versiones ligeramente diferentes de los mismos datos. Por ejemplo:

- Los resultados de las consultas en caché suelen tolerar datos ligeramente obsoletos, especialmente en el caso de los patrones de reserva de caché, en los que el origen de la verdad es externo.
- En una tabla de clasificación de videojuegos, un retraso de unos segundos en la actualización de las puntuaciones no suele afectar de forma significativa a la experiencia del usuario.
- En el caso de los almacenes de sesiones, algunos retrasos leves en la propagación de los datos de sesión entre las réplicas rara vez afectan a la funcionalidad de la aplicación.
- Los motores de recomendaciones suelen utilizar el análisis de datos históricos, por lo que la coherencia en tiempo real es menos importante.

Coherencia final significa que todos los nodos de réplica devolverán los mismos datos una vez que se complete el proceso de replicación, normalmente en cuestión de milisegundos. En estos casos de uso, la implementación de réplicas de lectura es una estrategia eficaz para reducir la latencia al leer desde la instancia de ElastiCache.

El uso de réplicas de lectura en Amazon ElastiCache puede proporcionar importantes mejoras del rendimiento gracias a:

Mejora de la escalabilidad de lectura

- Distribuye las operaciones de lectura entre varios nodos de réplica.
- Las descargas leen el tráfico de lectura del nodo principal.
- Se reduce la latencia de lectura al atender las solicitudes de réplicas más cercanas desde el punto de vista geográfico.

Rendimiento optimizado del nodo principal

- Asigna los recursos del nodo principal a las operaciones de escritura.
- Reduce la sobrecarga de la conexión en el nodo principal.
- Mejora el rendimiento de escritura y mantiene mejores tiempos de respuesta durante los períodos de más tráfico.

Uso de la lectura desde la réplica en ElastiCache sin servidor

ElastiCache sin servidor proporciona dos puntos de conexión diferentes para distintos requisitos de coherencia. Los dos puntos de conexión utilizan el mismo nombre de DNS pero puertos diferentes.

Para utilizar el puerto de lectura desde la réplica, debe autorizar el acceso a ambos puertos desde la aplicación cliente mediante la configuración de los grupos de seguridad y las listas de control de acceso a la red de su VPC.

Punto de conexión principal (puerto 6379)

- Se utiliza para operaciones que requieren coherencia inmediata.
- Garantiza que se lean los datos más actualizados.
- Es ideal para transacciones y operaciones de escritura críticas.
- Es necesario para las operaciones de escritura.
- Ejemplo: `test-12345.serverless.use1.cache.amazonaws.com:6379`

Punto de conexión con latencia optimizada (puerto 6380)

- Optimizado para operaciones de lectura que pueden tolerar una coherencia final.
- Si se puede, ElastiCache sin servidor enruta automáticamente las solicitudes de lectura al nodo de réplica en la zona de disponibilidad local del cliente. Esta optimización proporciona una latencia más baja al evitar la latencia de red adicional que se produce al recuperar datos de un nodo en una zona de disponibilidad diferente.
- ElastiCache sin servidor selecciona automáticamente los nodos disponibles en otras zonas si un nodo local no está disponible.
- Ejemplo: `test-12345.serverless.use1.cache.amazonaws.com:6380`
- Los clientes como Glide y Lettuce detectarán automáticamente las lecturas y las enrutarán hacia el punto de conexión con latencia optimizada si proporciona la configuración de lectura desde la réplica. Si su cliente no admite la configuración de enrutamiento (por ejemplo, valkey-java y versiones anteriores de jedis), debe definir la configuración de puerto y cliente correcta para leer las réplicas.

Conexión a réplicas de lectura en ElastiCache sin servidor: Valkey y Glide

El siguiente fragmento de código muestra cómo se puede configurar la lectura desde la réplica para ElastiCache sin servidor en la biblioteca Valkey Glide. No necesita especificar el puerto para la lectura de las réplicas, pero sí configurar el enrutamiento `ReadFrom.PREFER_REPLICA`.

```
package glide.examples;
```

```
import glide.api.GlideClusterClient;
import glide.api.logging.Logger;
import glide.api.models.configuration.GlideClusterClientConfiguration;
import glide.api.models.configuration.NodeAddress;
import glide.api.models.exceptions.ClosingException;
import glide.api.models.exceptions.ConnectionException;
import glide.api.models.exceptions.TimeoutException;
import glide.api.models.configuration.ReadFrom;

import java.util.concurrent.CompletableFuture;
import java.util.concurrent.ExecutionException;

public class ClusterExample {

    public static void main(String[] args) {
        // Set logger configuration
        Logger.setLoggerConfig(Logger.Level.INFO);

        GlideClusterClient client = null;

        try {
            System.out.println("Connecting to Valkey Glide...");

            // Configure the Glide Client
            GlideClusterClientConfiguration config =
 GlideClusterClientConfiguration.builder()
                .address(NodeAddress.builder()
                    .host("your-endpoint")
                    .port(6379)
                    .build())
                .useTLS(true)
                .readFrom(ReadFrom.PREFER_REPLICA)
                .build();

            // Create the GlideClusterClient
            client = GlideClusterClient.createClient(config).get();
            System.out.println("Connected successfully.");

            // Perform SET operation
            CompletableFuture<String> setResponse = client.set("key", "value");
            System.out.println("Set key 'key' to 'value': " + setResponse.get());

            // Perform GET operation
            CompletableFuture<String> getResponse = client.get("key");
```

```
            System.out.println("Get response for 'key': " + getResponse.get());

            // Perform PING operation
            CompletableFuture<String> pingResponse = client.ping();
            System.out.println("PING response: " + pingResponse.get());

        } catch (ClosingException | ConnectionException | TimeoutException |
 ExecutionException e) {
            System.err.println("An exception occurred: ");
            e.printStackTrace();
        } catch (InterruptedException e) {
            Thread.currentThread().interrupt();
        } finally {
            // Close the client connection
            if (client != null) {
                try {
                    client.close();
                    System.out.println("Client connection closed.");
                } catch (ClosingException | ExecutionException e) {
                    System.err.println("Error closing client: " + e.getMessage());
                }
            }
        }
    }
}
```

Comandos de Valkey, Memcached y Redis OSS compatibles y restringidos

Comandos de Valkey y Redis OSS compatibles

Comandos de Valkey y Redis OSS compatibles

Las cachés sin servidor admiten los siguientes comandos de Valkey y Redis OSS. Además de estos comandos, estos Comandos de Valkey y Redis OSS compatibles también son compatibles.

Para obtener más información sobre los comandos del filtro de Bloom, consulte Comandos de filtros de Bloom

Comandos de mapa de bits

- `BITCOUNT`

  Cuenta el número de bits establecidos (recuento de integrantes) de una cadena.

Más información

- `BITFIELD`

  Realiza operaciones arbitrarias con enteros de campos de bits en cadenas.

  Más información

- `BITFIELD_RO`

  Realiza operaciones arbitrarias de solo lectura con enteros de campos de bits en cadenas.

  Más información

- `BITOP`

  Realiza operaciones bit a bit en varias cadenas y almacena el resultado.

  Más información

- `BITPOS`

  Busca el primer bit activo (1) o inactivo (0) en una cadena.

  Más información

- `GETBIT`

  Devuelve el valor de un bit en una posición determinada.

  Más información

- `SETBIT`

  Pone a 1 o 0 el bit de una posición determinada en el valor de la cadena. Crea la clave si esta no existe.

  Más información

Comandos de administración de clústeres

- `CLUSTER COUNTKEYSINSLOT`

  Devuelve el número de claves en un slot hash.

Más información

- `CLUSTER GETKEYSINSLOT`

  Devuelve los nombres de las claves de un slot hash.

  Más información

- `CLUSTER INFO`

  Devuelve información sobre el estado de un nodo. En una caché sin servidor, devuelve el estado de la única “partición” virtual expuesta al cliente.

  Más información

- `CLUSTER KEYSLOT`

  Devuelve el slot hash de una clave.

  Más información

- `CLUSTER MYID`

  Devuelve el ID de un nodo. En una caché sin servidor, devuelve el estado de la única “partición” virtual expuesta al cliente.

  Más información

- `CLUSTER NODES`

  Devuelve la configuración del clúster de un nodo. En una caché sin servidor, devuelve el estado de la única “partición” virtual expuesta al cliente.

  Más información

- `CLUSTER REPLICAS`

  Muestra los nodos de réplica de un nodo maestro. En una caché sin servidor, devuelve el estado de la única “partición” virtual expuesta al cliente.

  Más información

- `CLUSTER SHARDS`

  Devuelve la asignación de slots del clúster a las particiones. En una caché sin servidor, devuelve el estado de la única “partición” virtual expuesta al cliente.

Más información

- `CLUSTER SLOTS`

  Devuelve la asignación de slots del clúster a los nodos. En una caché sin servidor, devuelve el estado de la única “partición” virtual expuesta al cliente.

  Más información

- `CLUSTER SLOT-STATS`

  Permite realizar un seguimiento de las métricas por ranura del recuento de claves, el uso de la CPU, los bytes de red entrantes y los bytes de red salientes.

  Más información

- `READONLY`

  Habilita las consultas de solo lectura para una conexión a un nodo de réplica de un clúster de Valkey o Redis OSS.

  Más información

- `READWRITE`

  Habilita las consultas de lectura-escritura para una conexión a un nodo de réplica de un clúster de Valkey o Redis OSS.

  Más información

- `SCRIPT SHOW`

  Devuelve el código fuente original de un script en la caché de scripts.

  Más información

Comandos de administración de conexiones

- `AUTH`

  Autentica la conexión.

  Más información

- `CLIENT GETNAME`

Devuelve el nombre de la conexión.

Más información

- `CLIENT REPLY`

  Le dice al servidor si debe responder a los comandos.

  Más información

- `CLIENT SETNAME`

  Establece el nombre de la conexión.

  Más información

- `ECHO`

  Devuelve la cadena determinada.

  Más información

- `HELLO`

  Protocolos de enlace con el servidor de Valkey o Redis OSS.

  Más información

- `PING`

  Devuelve la respuesta de vivacidad del servidor.

  Más información

- `QUIT`

  Cierra la conexión.

  Más información

- `RESET`

  Restablece la conexión.

  Más información

- `SELECT`

Cambia la base de datos seleccionada.

Más información

Comandos genéricos

- `COPY`

  Copia el valor de una clave en una nueva clave.

  Más información
- `DEL`

  Elimina una o varias claves.

  Más información
- `DUMP`

  Devuelve una representación serializada del valor almacenado en una clave.

  Más información
- `EXISTS`

  Determina si hay una o más claves.

  Más información
- `EXPIRE`

  Establece el tiempo de caducidad de una clave en segundos.

  Más información
- `EXPIREAT`

  Establece el tiempo de caducidad de una clave en una marca de tiempo de Unix.

  Más información
- `EXPIRETIME`

  Devuelve el tiempo de caducidad de una clave como una marca de tiempo de Unix.

Más información

- `PERSIST`

  Elimina el tiempo de caducidad de una clave.

  Más información

- `PEXPIRE`

  Establece el tiempo de caducidad de una clave en segundos.

  Más información

- `PEXPIREAT`

  Establece el tiempo de caducidad de una clave como una marca de milisegundos de Unix.

  Más información

- `PEXPIRETIME`

  Devuelve el tiempo de caducidad de una clave como una marca de tiempo de milisegundos de Unix.

  Más información

- `PTTL`

  Devuelve el tiempo de caducidad de una clave en milisegundos.

  Más información

- `RANDOMKEY`

  Devuelve un nombre de clave aleatorio de la base de datos.

  Más información

- `RENAME`

  Cambia el nombre de una clave y sobrescribe el destino.

  Más información

- `RENAMENX`

Cambia el nombre de una clave solo cuando el nombre de la clave de destino no existe.

Más información

- `RESTORE`

  Crea una clave a partir de la representación serializada de un valor.

  Más información

- `SCAN`

  Hace una iteración sobre los nombres de claves en la base de datos.

  Más información

- `SORT`

  Ordena los elementos de una lista, un conjunto o un conjunto ordenado y, opcionalmente, almacena el resultado.

  Más información

- `SORT_RO`

  Devuelve los elementos ordenados de una lista, un conjunto o un conjunto ordenado.

  Más información

- `TOUCH`

  Devuelve el número de claves existentes entre las claves especificadas tras actualizar la hora en la que se accedió a ellas por última vez.

  Más información

- `TTL`

  Devuelve el tiempo de caducidad de una clave en segundos.

  Más información

- `TYPE`

  Determina el tipo de valor almacenado en una clave.

Más información

- UNLINK

  Elimina de forma asíncrona una o más claves.

  Más información

Comandos geoespaciales

- GEOADD

  Añade uno o varios miembros a un índice geoespacial. La clave se crea si no existe.

  Más información

- GEODIST

  Devuelve la distancia entre dos miembros de un índice geoespacial.

  Más información

- GEOHASH

  Devuelve los miembros de un índice geoespacial como cadenas geohash.

  Más información

- GEOPOS

  Devuelve la longitud y la latitud de los miembros de un índice geoespacial.

  Más información

- GEORADIUS

  Consulta un índice geoespacial para ver los miembros que se encuentran a una distancia de una coordenada y, si lo desea, almacena el resultado.

  Más información

- GEORADIUS_RO

  Devuelve los miembros de un índice geoespacial que se encuentran a una determinada distancia de una coordenada.

Más información

- `GEORADIUSBYMEMBER`

  Consulta un índice geoespacial para ver los miembros que se encuentran a una determinada distancia de un miembro y, si lo desea, almacena el resultado.

  Más información

- `GEORADIUSBYMEMBER_RO`

  Devuelve los miembros de un índice geoespacial que se encuentran a una determinada distancia de un miembro.

  Más información

- `GEOSEARCH`

  Consulta en un índice geoespacial a los miembros que se encuentran dentro del área de un cuadro o un círculo.

  Más información

- `GEOSEARCHSTORE`

  Consulta en un índice geoespacial a los miembros que se encuentran dentro del área de un cuadro o un círculo y, opcionalmente, almacena el resultado.

  Más información

Comandos hash

- `HDEL`

  Elimina uno o más campos (y sus valores) de un hash. Elimina el hash si no queda ningún campo.

  Más información

- `HEXISTS`

  Determina si hay un campo en un hash.

  Más información

- `HGET`

Devuelve el valor de un campo en un hash.

Más información

- `HGETALL`

  Devuelve todos los campos y valores de un hash.

  Más información

- `HINCRBY`

  Incrementa el valor entero de un campo en un hash por un número determinado. Usa 0 como valor inicial si el campo no existe.

  Más información

- `HINCRBYFLOAT`

  Incrementa el valor de punto flotante de un campo por un número determinado. Usa 0 como valor inicial si el campo no existe.

  Más información

- `HKEYS`

  Devuelve todos los campos de un hash.

  Más información

- `HLEN`

  Devuelve el número de campos en un hash.

  Más información

- `HMGET`

  Devuelve todos los campos y valores en un hash.

  Más información

- `HMSET`

  Establece los valores de varios campos.

Más información

- HRANDFIELD

  Devuelve uno o más campos aleatorios de un hash.

  Más información

- HSCAN

  Hace iteraciones sobre los campos y valores de un hash.

  Más información

- HSET

  Crea o modifica el valor de un campo en un hash.

  Más información

- HSETNX

  Establece el valor de un campo en un hash solo cuando el campo no existe.

  Más información

- HSTRLEN

  Devuelve la longitud del valor de un campo.

  Más información

- HVALS

  Devuelve todos valores de un hash.

  Más información

Comandos HyperLogLog

- PFADD

  Añade elementos a una clave de HyperLogLog. Crea la clave si esta no existe.

  Más información

- PFCOUNT

  Devuelve la cardinalidad aproximada de los conjuntos observados por las claves de HyperLogLog.

  Más información

- PFMERGE

  Combina uno o más valores de HyperLogLog en una sola clave.

  Más información

Comandos de listas

- BLMOVE

  Saca un elemento de una lista, lo coloca en otra y lo devuelve. De lo contrario, bloquea hasta que haya un elemento disponible. Elimina la lista si se ha movido el último elemento.

  Más información

- BLMPOP

  Saca el primer elemento de una de las múltiples listas. De lo contrario, bloquea hasta que haya un elemento disponible. Elimina la lista si se ha sacado el último elemento.

  Más información

- BLPOP

  Elimina y devuelve el primer elemento de una lista. De lo contrario, bloquea hasta que haya un elemento disponible. Elimina la lista si se ha sacado el último elemento.

  Más información

- BRPOP

  Elimina y devuelve el último elemento de una lista. De lo contrario, bloquea hasta que haya un elemento disponible. Elimina la lista si se ha sacado el último elemento.

  Más información

- BRPOPLPUSH

Saca un elemento de una lista, lo coloca en otra y lo devuelve. De lo contrario, bloquea hasta que haya un elemento disponible. Elimina la lista si se ha sacado el último elemento.

Más información

- `LINDEX`

  Devuelve un elemento de una lista por su índice.

  Más información

- `LINSERT`

  Inserta un elemento antes o después de otro elemento de una lista.

  Más información

- `LLEN`

  Devuelve la longitud de una lista.

  Más información

- `LMOVE`

  Devuelve un elemento después de sacarlo de una lista y pasarlo a otra. Elimina la lista si se ha movido el último elemento.

  Más información

- `LMPOP`

  Devuelve varios elementos de una lista después de eliminarlos. Elimina la lista si se ha sacado el último elemento.

  Más información

- `LPOP`

  Devuelve los primeros elementos de una lista después de eliminarla. Elimina la lista si se ha sacado el último elemento.

  Más información

- `LPOS`

Devuelve el índice de los elementos coincidentes de una lista.

Más información

- LPUSH

  Antepone uno o más elementos a una lista. Crea la clave si esta no existe.

  Más información

- LPUSHX

  Antepone uno o más elementos a una lista solo cuando la lista existe.

  Más información

- LRANGE

  Devuelve un rango de elementos de una lista.

  Más información

- LREM

  Elimina elementos de una lista. Elimina la lista si se ha eliminado el último elemento.

  Más información

- LSET

  Establece el valor de un elemento de una lista por su índice.

  Más información

- LTRIM

  Elimina los elementos de ambos extremos de la lista. Elimina la lista si se han recortado todos los elementos.

  Más información

- RPOP

  Elimina y devuelve los últimos elementos de una lista. Elimina la lista si se ha sacado el último elemento.

Más información

- `RPOPLPUSH`

  Devuelve el último elemento de una lista después de extraerlo y colocarlo en otra lista. Elimina la lista si se ha sacado el último elemento.

  Más información

- `RPUSH`

  Antepone uno o más elementos a una lista. Crea la clave si esta no existe.

  Más información

- `RPUSHX`

  Añade un elemento a una lista solo cuando la lista existe.

  Más información

Comandos Pub/Sub

> **Note**
>
> Los comandos PUBSUB utilizan internamente PUBSUB con particiones, por lo que los nombres de los canales se mezclarán.

- `PUBLISH`

  Publica un mensaje en un canal.

  Más información

- `PUBSUB CHANNELS`

  Devuelve los canales activos.

  Más información

- `PUBSUB NUMSUB`

  Devuelve el recuento de suscriptores a los canales.

Más información

- `PUBSUB SHARDCHANNELS`

  Devuelve los canales de particiones activos.

  Más información

- `PUBSUB SHARDNUMSUB`

  Devuelve el recuento de suscriptores a los canales de particiones.

  Más información

- `SPUBLISH`

  Publicación de un mensaje en un canal de partición

  Más información

- `SSUBSCRIBE`

  Escucha los mensajes publicados en los canales de particiones.

  Más información

- `SUBSCRIBE`

  Escucha los mensajes publicados en los canales.

  Más información

- `SUNSUBSCRIBE`

  Deja de escuchar los mensajes publicados en los canales de particiones.

  Más información

- `UNSUBSCRIBE`

  Deja de escuchar los mensajes publicados en los canales.

  Más información

Comandos de scripts

- `EVAL`

  Ejecuta un script de Lua en el lado del servidor.

  Más información

- `EVAL_RO`

  Ejecuta un script de Lua de solo lectura en el lado del servidor.

  Más información

- `EVALSHA`

  Ejecuta un script de Lua en el lado del servidor con el identificador SHA1 digest.

  Más información

- `EVALSHA_RO`

  Ejecuta un script de Lua de solo lectura en el lado del servidor con el identificador SHA1 digest.

  Más información

- `SCRIPT EXISTS`

  Determina si hay scripts de Lua del lado del servidor en la caché de scripts.

  Más información

- `SCRIPT FLUSH`

  Actualmente no está en funcionamiento y la caché del script está administrada por el servicio.

  Más información

- `SCRIPT LOAD`

  Carga un script de Lua del lado del servidor en la caché de scripts.

  Más información

Comandos de administración de servidores

> **Note**
>
> Cuando se utilizan clústeres de ElastiCache basados en nodos para Valkey y Redis OSS, el cliente debe enviar los comandos de vaciado a todos los clústeres principales para vaciar todas las claves. ElastiCache sin servidor para Valkey y Redis OSS funciona de forma diferente, ya que abstrae la topología del clúster subyacente. ElastiCache sin servidor, el resultado es que los comandos `FLUSHDB` y `FLUSHALL` siempre vaciarán todas las claves del clúster. Por este motivo, los comandos de vaciado no se pueden incluir en una transacción sin servidor.

- `ACL CAT`

  Muestra las categorías de ACL o los comandos dentro de una categoría.

  Más información
- `ACL GENPASS`

  Genera una contraseña segura y pseudoaleatoria que se puede utilizar para identificar a los usuarios de ACL.

  Más información
- `ACL GETUSER`

  Muestra las reglas de ACL de un usuario.

  Más información
- `ACL LIST`

  Coloca las reglas vigentes en formato de archivo ACL.

  Más información
- `ACL USERS`

  Muestra todos los usuarios de ACL.

  Más información
- `ACL WHOAMI`

Devuelve el nombre de usuario autenticado de la conexión actual.

Más información

- `DBSIZE`

  Devuelve el número de claves de la base de datos actualmente seleccionada. No se garantiza que esta operación sea integral en todos los slots.

  Más información

- `COMMAND`

  Devuelve información detallada sobre todos los comandos.

  Más información

- `COMMAND COUNT`

  Devuelve un recuento de comandos.

  Más información

- `COMMAND DOCS`

  Devuelve información documental sobre un comando, varios o todos ellos.

  Más información

- `COMMAND GETKEYS`

  Extrae los nombres de claves de un comando arbitrario.

  Más información

- `COMMAND GETKEYSANDFLAGS`

  Extrae los nombres de claves y los indicadores de acceso de un comando arbitrario.

  Más información

- `COMMAND INFO`

  Devuelve información sobre un comando, varios o todos ellos.

  Más información

- `COMMAND LIST`

  Devuelve una lista de nombres de comandos.

  Más información

- `COMMANDLOG`

  Contenedor para los comandos del registro de comandos.

  Más información

- `COMMANDLOG GET`

  Devuelve las entradas del registro de comandos especificado.

  Más información

- `COMMANDLOG HELP`

  Muestra texto útil sobre los distintos subcomandos.

  Más información

- `COMMANDLOG LEN`

  Devuelve el número de entradas del tipo de registro de comandos especificado.

  Más información

- `COMMANDLOG RESET`

  Borra todas las entradas del tipo de registro de comandos especificado.

  Más información

- `FLUSHALL`

  Elimina todas las claves de todas las bases de datos. No se garantiza que esta operación sea integral en todos los slots.

  Más información

- `FLUSHDB`

  Elimina todas las claves de la base de datos actual. No se garantiza que esta operación sea integral en todos los slots.

Más información

- `INFO`

  Devuelve información y estadísticas sobre el servidor.

  Más información

- `LOLWUT`

  Muestra expresiones de arte computacional y la versión de Valkey o Redis OSS.

  Más información

- `ROLE`

  Devuelve la función de réplica.

  Más información

- `TIME`

  Devuelve la hora del servidor.

  Más información

Comandos para conjuntos

- `SADD`

  Añade uno o varios miembros a un conjunto. Crea la clave si esta no existe.

  Más información

- `SCARD`

  Devuelve el número de miembros de un conjunto.

  Más información

- `SDIFF`

  Devuelve la diferencia de varios conjuntos.

  Más información

- `SDIFFSTORE`

  Almacena la diferencia de varios conjuntos en una clave.

  Más información

- `SINTER`

  Devuelve la intersección de varios conjuntos.

  Más información

- `SINTERCARD`

  Devuelve el número de miembros de la intersección de varios conjuntos.

  Más información

- `SINTERSTORE`

  Almacena la intersección de varios conjuntos en una clave.

  Más información

- `SISMEMBER`

  Determina si un miembro pertenece a un conjunto.

  Más información

- `SMEMBERS`

  Devuelve todos los miembros de un conjunto.

  Más información

- `SMISMEMBER`

  Determina si varios miembros pertenecen a un conjunto.

  Más información

- `SMOVE`

  Mueve un miembro de un conjunto a otro.

  Más información

- SPOP

  Devuelve uno o más miembros aleatorios de un conjunto después de eliminarlos. Elimina la lista si se ha sacado el último miembro.

  Más información

- SRANDMEMBER

  Obtiene uno o varios miembros aleatorios de un conjunto

  Más información

- SREM

  Elimina uno o varios miembros de un conjunto. Elimina el conjunto si se ha extraído el último miembro.

  Más información

- SSCAN

  Hace iteraciones sobre los miembros de un conjunto.

  Más información

- SUNION

  Devuelve la unión de varios conjuntos.

  Más información

- SUNIONSTORE

  Almacena la unión de varios conjuntos en una clave.

  Más información

Comandos de conjuntos ordenados

- BZMPOP

  Elimina y devuelve un miembro por puntuación desde uno o más conjuntos ordenados. De lo contrario, bloquea hasta que haya un miembro disponible. Elimina el conjunto ordenado si se ha sacado el último elemento.

Más información

- BZPOPMAX

  Elimina y devuelve el miembro con la puntuación más alta de uno o más conjuntos ordenados. De lo contrario, bloquea hasta que haya un miembro disponible. Elimina el conjunto ordenado si se ha sacado el último elemento.

  Más información

- BZPOPMIN

  Elimina y devuelve el miembro con la puntuación más baja de uno o más conjuntos ordenados. De lo contrario, bloquea hasta que haya un miembro disponible. Elimina el conjunto ordenado si se ha sacado el último elemento.

  Más información

- ZADD

  Añade uno o más miembros a un conjunto ordenado o actualiza sus puntuaciones. Crea la clave si esta no existe.

  Más información

- ZCARD

  Devuelve el número de miembros de un conjunto ordenado.

  Más información

- ZCOUNT

  Devuelve el recuento de miembros de un conjunto ordenado con puntuaciones dentro de un rango determinado.

  Más información

- ZDIFF

  Devuelve la diferencia entre varios conjuntos ordenados.

  Más información

- ZDIFFSTORE

Almacena la diferencia de varios conjuntos en una clave.

Más información

- `ZINCRBY`

  Incrementa la puntuación de un miembro en un conjunto ordenado.

  Más información

- `ZINTER`

  Devuelve la intersección de varios conjuntos ordenados.

  Más información

- `ZINTERCARD`

  Devuelve el número de miembros de la intersección de varios conjuntos ordenados.

  Más información

- `ZINTERSTORE`

  Almacena la intersección de varios conjuntos ordenados en una clave.

  Más información

- `ZLEXCOUNT`

  Devuelve el número de miembros de un conjunto ordenado dentro de un rango lexicográfico.

  Más información

- `ZMPOP`

  Devuelve los miembros con la puntuación más alta o más baja de uno o más conjuntos ordenados después de eliminarlos. Elimina el conjunto ordenado si se ha sacado el último miembro.

  Más información

- `ZMSCORE`

  Devuelve la puntuación de uno o más miembros de un conjunto ordenado.

  Más información

- ZPOPMAX

  Devuelve los miembros con la puntuación más alta de un conjunto ordenado después de eliminarlos. Elimina el conjunto ordenado si se ha sacado el último miembro.

  Más información

- ZPOPMIN

  Devuelve los miembros con la puntuación más baja de un conjunto ordenado después de eliminarlos. Elimina el conjunto ordenado si se ha sacado el último miembro.

  Más información

- ZRANDMEMBER

  Devuelve uno o más miembros aleatorios de un conjunto ordenado.

  Más información

- ZRANGE

  Devuelve los miembros de un conjunto ordenado dentro de un rango de índices.

  Más información

- ZRANGEBYLEX

  Devuelve los miembros de un conjunto ordenado dentro de un rango lexicográfico.

  Más información

- ZRANGEBYSCORE

  Devuelve los miembros de un conjunto ordenado dentro de un rango de puntuaciones.

  Más información

- ZRANGESTORE

  Almacena un rango de miembros de un conjunto ordenado en una clave.

  Más información

- ZRANK

  Devuelve el índice de un miembro en un conjunto ordenado por puntuaciones ascendentes.

Más información

- ZREM

  Elimina uno o varios miembros de un conjunto ordenado. Elimina el conjunto ordenado si se han eliminado todos los miembros.

  Más información

- ZREMRANGEBYLEX

  Elimina los miembros de un conjunto ordenado dentro de un rango lexicográfico. Elimina el conjunto ordenado si se han eliminado todos los miembros.

  Más información

- ZREMRANGEBYRANK

  Elimina los miembros de un conjunto ordenado dentro de un rango de índices. Elimina el conjunto ordenado si se han eliminado todos los miembros.

  Más información

- ZREMRANGEBYSCORE

  Elimina los miembros de un conjunto ordenado dentro de un rango de puntuaciones. Elimina el conjunto ordenado si se han eliminado todos los miembros.

  Más información

- ZREVRANGE

  Devuelve los miembros de un conjunto ordenado dentro de un rango de índices en orden inverso.

  Más información

- ZREVRANGEBYLEX

  Devuelve los miembros de un conjunto ordenado dentro de un rango lexicográfico en orden inverso.

  Más información

- ZREVRANGEBYSCORE

Devuelve los miembros de un conjunto ordenado dentro de un rango de puntuaciones en orden inverso.

Más información

- ZREVRANK

  Devuelve el índice de un miembro en un conjunto ordenado por puntuaciones descendientes.

  Más información

- ZSCAN

  Hace iteraciones sobre los miembros y las puntuaciones de un conjunto ordenado.

  Más información

- ZSCORE

  Devuelve la puntuación de un miembro en un conjunto ordenado.

  Más información

- ZUNION

  Devuelve la unión de varios conjuntos ordenados.

  Más información

- ZUNIONSTORE

  Almacena la unión de varios conjuntos ordenados en una clave.

  Más información

Comandos de transmisión

- XACK

  Devuelve el número de mensajes confirmados correctamente por el miembro del grupo de consumidores de una transmisión.

  Más información

- XADD

Añade un mensaje nuevo a una transmisión. Crea la clave si esta no existe.

Más información

- `XAUTOCLAIM`

  Cambia o adquiere la propiedad de los mensajes de un grupo de consumidores, como si los mensajes se hubieran entregado como miembro del grupo de consumidores.

  Más información

- `XCLAIM`

  Cambia o adquiere la propiedad de un mensaje en un grupo de consumidores, como si el mensaje se hubiera entregado por parte de un miembro del grupo de consumidores.

  Más información

- `XDEL`

  Devuelve el número de mensajes después de eliminarlos de una transmisión.

  Más información

- `XGROUP CREATE`

  Crea un grupo de consumidores.

  Más información

- `XGROUP CREATECONSUMER`

  Crea un consumidor en un grupo de consumidores.

  Más información

- `XGROUP DELCONSUMER`

  Elimina un consumidor de un grupo de consumidores.

  Más información

- `XGROUP DESTROY`

  Destruye un grupo de consumidores.

Más información

- `XGROUP SETID`

  Establece el último ID entregado de un grupo de consumidores.

  Más información

- `XINFO CONSUMERS`

  Devuelve una lista de los consumidores de un grupo de consumidores.

  Más información

- `XINFO GROUPS`

  Devuelve una lista de los grupos de consumidores de una transmisión.

  Más información

- `XINFO STREAM`

  Devuelve información sobre una transmisión.

  Más información

- `XLEN`

  Devuelve la cantidad de mensajes en una transmisión.

  Más información

- `XPENDING`

  Devuelve la información y las entradas de la lista de entradas pendientes de un grupo de consumidores de una transmisión.

  Más información

- `XRANGE`

  Devuelve los mensajes de una transmisión dentro de un rango de ID.

  Más información

- `XREAD`

Devuelve los mensajes de varias transmisiones con ID superiores a los solicitados. De lo contrario, bloquea hasta que haya un mensaje disponible.

Más información

- XREADGROUP

  Devuelve mensajes nuevos o históricos de una transmisión para un consumidor en un grupo. De lo contrario, bloquea hasta que haya un mensaje disponible.

  Más información

- XREVRANGE

  Devuelve los mensajes de una transmisión dentro de un rango de ID en orden inverso.

  Más información

- XTRIM

  Elimina los mensajes desde el principio de una transmisión.

  Más información

Comandos de cadenas

- APPEND

  Añade una cadena al valor de una clave. Crea la clave si esta no existe.

  Más información

- DECR

  Disminuye en uno el valor entero de una clave. Utiliza 0 como valor inicial si la clave no existe.

  Más información

- DECRBY

  Disminuye un número del valor entero de una clave. Utiliza 0 como valor inicial si la clave no existe.

  Más información

- `GET`

  Devuelve el valor de cadena de una clave.

  Más información

- `GETDEL`

  Devuelve el valor de cadena de una clave tras eliminarla.

  Más información

- `GETEX`

  Devuelve el valor de cadena de una clave tras establecer su fecha de caducidad.

  Más información

- `GETRANGE`

  Devuelve una subcadena de la cadena almacenada en una clave.

  Más información

- `GETSET`

  Devuelve el valor de cadena anterior de una clave después de establecerla en un nuevo valor.

  Más información

- `INCR`

  Aumenta en uno el valor entero de una clave. Utiliza 0 como valor inicial si la clave no existe.

  Más información

- `INCRBY`

  Incrementa en un número determinado el valor entero de una clave. Utiliza 0 como valor inicial si la clave no existe.

  Más información

- `INCRBYFLOAT`

  Incrementa en un número determinado el valor de punto flotante de una clave. Utiliza 0 como valor inicial si la clave no existe.

Más información

- `LCS`

  Busca la subcadena común más larga.

  Más información

- `MGET`

  Devuelve íntegramente los valores de cadena de una o más claves.

  Más información

- `MSET`

  Crea o modifica íntegramente los valores de cadena de una o más claves.

  Más información

- `MSETNX`

  Modifica íntegramente los valores de cadena de una o más claves solo cuando no existen todas las claves.

  Más información

- `PSETEX`

  Establece tanto el valor de la cadena como el tiempo de caducidad en milisegundos de una clave. La clave se crea si no existe.

  Más información

- `SET`

  Establece el valor de cadena de una clave e ignora el tipo. La clave se crea si no existe.

  Más información

- `SETEX`

  Establece el valor de la cadena y el tiempo de caducidad de una clave. Crea la clave si esta no existe.

  Más información

- `SETNX`

  Establece el valor de cadena de una clave solo cuando la clave no existe.

  Más información

- `SETRANGE`

  Sobrescribe una parte del valor de una cadena por otra a partir de una determinada posición. Crea la clave si esta no existe.

  Más información

- `STRLEN`

  Devuelve la longitud del valor de la cadena.

  Más información

- `SUBSTR`

  Devuelve una subcadena a partir de un valor de cadena.

  Más información

Comandos de transacción

- `DISCARD`

  Descarta una transacción.

  Más información

- `EXEC`

  Ejecuta todos los comandos de una transacción.

  Más información

- `MULTI`

  Inicia una transacción.

  Más información

Comandos restringidos de Valkey y Redis OSS

Para ofrecer una experiencia de servicio administrado, ElastiCache restringe el acceso a determinados comandos específicos del motor de caché que requieren privilegios avanzados. En las cachés que ejecutan Redis OSS, los siguientes comandos no están disponibles:

- `acl setuser`
- `acl load`
- `acl save`
- `acl deluser`
- `bgrewriteaof`
- `bgsave`
- `cluster addslot`
- `cluster addslotsrange`
- `cluster bumpepoch`
- `cluster delslot`
- `cluster delslotsrange`
- `cluster failover`
- `cluster flushslots`
- `cluster forget`
- `cluster links`
- `cluster meet`
- `cluster setslot`
- `config`
- `debug`
- `migrate`
- `psync`
- `replicaof`
- `save`
- `slaveof`
- `shutdown`
- `sync`

Además, los siguientes comandos no están disponibles para las cachés sin servidor:

- acl log
- client caching
- client getredir
- client id
- client info
- client kill
- client list
- client no-evict
- client pause
- client tracking
- client trackinginfo
- client unblock
- client unpause
- cluster count-failure-reports
- commandlog
- commandlog get
- commandlog help
- commandlog len
- commandlog reset
- fcall
- fcall_ro
- function
- function delete
- function dump
- function flush
- function help
- function kill
- function list

- function load
- function restore
- function stats
- keys
- lastsave
- latency
- latency doctor
- latency graph
- latency help
- latency histogram
- latency history
- latency latest
- latency reset
- memory
- memory doctor
- memory help
- memory malloc-stats
- memory purge
- memory stats
- memory usage
- monitor
- move
- object
- object encoding
- object freq
- object help
- object idletime
- object refcount
- pfdebug
- pfselftest

- `psubscribe`
- `pubsub numpat`
- `punsubscribe`
- `script kill`
- `slowlog`
- `slowlog get`
- `slowlog help`
- `slowlog len`
- `slowlog reset`
- `swapdb`
- `unwatch`
- `wait`
- `watch`

Comandos compatibles con Memcached

ElastiCache Serverless para Memcached admite todos los comandos de memcached de la versión 1.6 de código abierto de memcached, con las siguientes excepciones:

- Las conexiones de cliente requieren TLS, por lo que no se admite el protocolo UDP.
- El protocolo binario no es compatible, ya que está oficialmente obsoleto en memcached 1.6.
- Los comandos `GET/GETS` están limitados a 16 KB para evitar posibles ataques de DoS al servidor con la obtención de una gran cantidad de claves.
- El comando `flush_all` retrasado se rechazará con `CLIENT_ERROR`.
- No se admiten los comandos que configuran el motor o revelan información interna sobre el estado o los registros del motor, como los siguientes:
  - En el comando `STATS`, solo se admiten `stats` y `stats reset`. Otras variantes devolverán un `ERROR`
  - `lru / lru_crawler`: modificación de la configuración de LRU y el rastreador de LRU
  - `watch`: observa los registros del servidor memcached
  - `verbosity`: configura el nivel de registro del servidor
  - `me`: no se admite el comando meta debug (me)

Configuración y límites de Valkey y Redis OSS

Los motores de Valkey y Redis OSS ofrecen una serie de parámetros de configuración; algunos de ellos se pueden modificar en ElastiCache para Redis OSS y otros no, para proporcionar un rendimiento y una fiabilidad estables.

Cachés sin servidor

En el caso de las cachés sin servidor, no se utilizan grupos de parámetros y no se puede modificar ninguna parte de la configuración de Valkey o Redis OSS. Los siguientes parámetros de Valkey o Redis OSS están implementados:

Nombre	Detalles	Descripción
`acl-pubsub-default`	`allchannels`	Permisos de canal pubsub predeterminados para los usuarios de ACL en la caché.

`client-output-buffer-limit`	`normal 0 0 0`
	
	`pubsub 32mb 8mb 60`
		Los clientes normales no tienen límite de búfer. Los clientes de PUB/SUB se desconectarán si superan una acumulación de 32 MiB o una acumulación de 8 MiB durante 60 segundos.

`client-query-buffer-limit`	1 GiB	El tamaño máximo de un búfer de consulta de cliente. Además, los clientes no pueden emitir una solicitud con más de 3999 argumentos.
`cluster-allow-pubsubshard-when-down`	`yes`	Esto permite que la caché sirva el tráfico de pubsub mientras esté parcialmente inactiva.
`cluster-allow-reads-when-down`	`yes`	Esto permite que la caché sirva el tráfico de lectura mientras esté parcialmente inactiva.
`cluster-enabled`	`yes`	Todas las cachés sin servidor están habilitadas para el modo de clúster, lo que les permite

Nombre	Detalles	Descripción
		dividir sus datos de forma transparente en varias particiones de backend. Los clientes observan todos los slots como si perteneciesen a un único nodo virtual.
`cluster-require-full-coverage`	no	Cuando el espacio de claves esté parcialmente inactivo (es decir, hay, al menos, un slot hash inaccesible), la memoria caché seguirá aceptando consultas para la parte del espacio de claves que aún esté cubierta. El espacio de claves en su totalidad estará siempre “cubierto” por un único nodo virtual en `cluster slots`.
`lua-time-limit`	5000	Tiempo de ejecución máximo para un script Lua, en milisegundos, antes de que ElastiCache emprenda acciones para detener el script. Si se supera el valor de `lua-time-limit` , todos los comandos de Valkey o Redis OSS devolverán un error con el formato ____-BUSY . Dado que este estado puede interferir con muchas operaciones esenciales de Valkey o Redis OSS, ElastiCache emitirá primero el comando SCRIPT KILL. Si esta operación no se realiza correctamente, ElastiCache forzará el reinicio de Valkey o Redis OSS.
`maxclients`	65 000	Número máximo de clientes que pueden conectarse a la vez. Las conexiones adicionales que se establezcan pueden realizarse correctamente o no.
`maxmemory-policy`	`volatile-lru`	Cuando se alcanza el límite de memoria de la caché, los elementos con un conjunto TTL son expulsados tras una estimación de uso menos reciente (LRU).

Nombre	Detalles	Descripción
`notify-keyspace-events`	(una cadena vacía)	Actualmente, los eventos de espacio de claves no son compatibles con las cachés sin servidor.
`port`	Puerto principal: 6379 Puerto de lectura: 6380	Las cachés sin servidor muestran dos puertos con el mismo nombre de host. El puerto principal permite escribir y leer, mientras que el puerto de lectura permite lecturas coherentes posteriores de menor latencia mediante el comando `READONLY`.
`proto-max-bulk-len`	512 MiB	El tamaño máximo de una sola solicitud de elemento.
`timeout`	0	Los clientes no se desconectan de forma ineludible tras un tiempo de inactividad específico, pero es posible que se desconect en durante el estado estable para equilibrar la carga.

Además, existen los siguientes límites:

Nombre	Detalles	Descripción
Tamaño por caché	5000 GiB	Cantidad máxima de datos que se puede almacenar por caché sin servidor.
Tamaño por slot	32 GiB	El tamaño máximo de una única ranura de hash de Valkey o Redis OSS. Los clientes que intenten almacenar más datos en una sola ranura de Valkey o Redis OSS activarán la política de expulsión en la ranura y, si no se puede expulsar ninguna clave, recibirán un error de memoria insuficiente (`OOM`).

Nombre	Detalles	Descripción
ECPU por caché	15 000 000 ECPU/segundo	Métrica de unidades de procesamiento de ElastiCache (ECPU). La cantidad de ECPU consumidas por las solicitudes depende del tiempo empleado por la vCPU y de la cantidad de datos transferidos.
ECPU por ranura	30 000 a 90 000 ECPU/segundo	Máximo de 30 000 ECPU/segundo por ranura o 90 000 ECPU/segundo cuando se utiliza Lectura desde réplica con conexiones READONLY.
Argumentos por solicitud	3999	Número máximo de argumentos por solicitud. Los clientes que envíen más argumentos por solicitud recibirán un error.
Longitud del nombre de la clave	4 KiB	El tamaño máximo de una sola clave de Valkey o Redis OSS o el nombre de un canal. Los clientes que hagan referencia a claves de un tamaño mayor a este valor recibirán un error.
El tamaño del script de Lua	4 MiB	El tamaño máximo de un único script de Lua de Valkey o Redis OSS. Los intentos de cargar un script de Lua con un tamaño superior a este recibirán un error.

Clústeres basados en nodos

Con respecto a los clústeres basados en nodos, consulte Parámetros de Valkey y Redis OSS para obtener información sobre los valores predeterminados de los parámetros de configuración y para saber cuáles son configurables. Por lo general, se recomiendan los valores predeterminados, a menos que tenga un caso de uso específico que requiera anularlos.

Ejemplos de clientes de IPv6 para Valkey, Memcached y Redis OSS

ElastiCache es compatible con Valkey, Memcached y Redis OSS. Esto significa que los clientes que admiten conexiones de IPv6 deberían poder conectarse a clústeres de ElastiCache para Memcached

habilitados para IPv6. Hay algunas advertencias que merece la pena señalar al interactuar con recursos habilitados para IPv6.

Puede consultar la publicación Best practices for Valkey and Redis clients en AWS Database Blog para obtener recomendaciones sobre la configuración de los clientes de Valkey y Redis OSS para los recursos de ElastiCache.

A continuación, se presentan las prácticas recomendadas para interactuar con los recursos de ElastiCache habilitados para IPv6 con las bibliotecas de cliente de código abierto que se utilizan habitualmente.

Clientes validados con Valkey y Redis OSS

ElastiCache es compatible con Valkey y Redis OSS de código abierto. Esto significa que los clientes de Valkey y Redis OSS de código abierto que admiten conexiones de IPv6 deberían poder conectarse a los clústeres de ElastiCache para Redis OSS habilitados para IPv6. Además, varios de los clientes de Python y Java más populares se han probado y validado específicamente para que funcionen con todas las configuraciones de tipos de red compatibles (solo IPv4, solo IPv6 y doble pila).

Los siguientes clientes se han validado específicamente para funcionar con todas las configuraciones de tipo de red admitidas para Valkey y Redis OSS.

Clientes validados:

- Redis Py (): 4.1.2
- Lettuce: versión 6.1.6.RELEASE
- Jedis: versión 3.6.0

Prácticas recomendadas para clientes (Valkey y Redis OSS)

Descubra las prácticas recomendadas en escenarios habituales y siga los ejemplos de código de algunas de las bibliotecas cliente de Valkey y Redis OSS de código abierto más populares (redis-py, PHPRedis y Lettuce), así como las prácticas recomendadas para interactuar con los recursos de ElastiCache con bibliotecas cliente de Memcached de código abierto de uso habitual.

Temas

- Gran cantidad de conexiones (Valkey y Redis OSS)
- Detección de clústeres por parte del cliente y retroceso exponencial (Valkey y Redis OSS)

- Configuración de un tiempo de espera del cliente (Valkey y Redis OSS)
- Configuración de un tiempo de espera del servidor (Valkey y Redis OSS)
- Scripts de Lua
- Almacenamiento de elementos compuestos de gran tamaño (Valkey y Redis OSS)
- Configuración del cliente de Lettuce (Valkey y Redis OSS)
- Configuración de un protocolo preferido para clústeres de doble pila (Valkey y Redis OSS)

Gran cantidad de conexiones (Valkey y Redis OSS)

Las cachés sin servidor y los nodos individuales de ElastiCache para Redis OSS admiten hasta 65 000 conexiones de clientes simultáneas. Sin embargo, para optimizar el rendimiento, recomendamos que las aplicaciones cliente no funcionen constantemente con ese volumen de conexiones. Valkey y Redis OSS tienen cada uno un subproceso basado en un bucle de eventos en el que las solicitudes entrantes de los clientes se gestionan de forma secuencial. Esto significa que el tiempo de respuesta de un cliente determinado se hace más largo a medida que aumenta el número de clientes conectados.

Puede tomar las siguientes medidas para evitar un cuello de botella en la conexión del servidor de Valkey o Redis OSS:

- Llevar a cabo las operaciones de lectura a partir de réplicas de lectura. Puede hacer esto usando los puntos de conexión del lector ElastiCache con el modo de clúster deshabilitado o usando réplicas para las lecturas con el modo de clúster habilitado, lo que incluye una caché sin servidor.
- Distribuir el tráfico de escritura entre varios nodos principales. Puede hacerlo de dos formas. Puede utilizar un clúster de Valkey y Redis OSS con varias particiones en un cliente compatible con el modo de clúster. También puede escribir en varios nodos principales con el modo de clúster deshabilitado y con partición en el lado del cliente. Esto se hace automáticamente en una memoria caché sin servidor.
- Usar un grupo de conexiones cuando esté disponible en la biblioteca de su cliente.

En general, crear una conexión de TCP es una operación costosa desde el punto de vista computacional en comparación con los comandos típicos de Valkey o Redis OSS. Por ejemplo, gestionar una solicitud SET/GET es una orden de magnitud más rápido cuando se reutiliza una conexión existente. El uso de un grupo de conexiones de clientes con un tamaño finito reduce la sobrecarga en la administración de conexiones. También limita el número de conexiones entrantes simultáneas desde la aplicación cliente.

El siguiente ejemplo de código de PHPRedis muestra cómo se crea una nueva conexión para cada nueva solicitud de usuario:

```
$redis = new Redis();
if ($redis->connect($HOST, $PORT) != TRUE) {
 //ERROR: connection failed
 return;
}
$redis->set($key, $value);
unset($redis);
$redis = NULL;
```

Hemos comparado este código en bucle en una instancia de Amazon Elastic Compute Cloud (Amazon EC2) conectada a un nodo ElastiCache para Redis OSS de Graviton2 (m6g.2xlarge). Hemos colocado el cliente y el servidor dentro de la misma zona de disponibilidad. La latencia media de toda la operación fue de 2,82 milisegundos.

Cuando actualizamos el código y utilizamos conexiones persistentes y un grupo de conexiones, la latencia media de toda la operación fue de 0,21 milisegundos:

```
$redis = new Redis();
if ($redis->pconnect($HOST, $PORT) != TRUE) {
 // ERROR: connection failed
 return;
}
$redis->set($key, $value);
unset($redis);
$redis = NULL;
```

Configuraciones de redis.ini obligatorias:

- `redis.pconnect.pooling_enabled=1`
- `redis.pconnect.connection_limit=10`

El siguiente código es un ejemplo de un grupo de conexiones Redis-py:

```
conn = Redis(connection_pool=redis.BlockingConnectionPool(host=HOST,
 max_connections=10))
conn.set(key, value)
```

El siguiente código es un ejemplo de un grupo de conexiones Lettuce:

```
RedisClient client = RedisClient.create(RedisURI.create(HOST, PORT));
GenericObjectPool<StatefulRedisConnection> pool =
 ConnectionPoolSupport.createGenericObjectPool(() -> client.connect(), new
 GenericObjectPoolConfig());
pool.setMaxTotal(10); // Configure max connections to 10
try (StatefulRedisConnection connection = pool.borrowObject()) {
 RedisCommands syncCommands = connection.sync();
 syncCommands.set(key, value);
}
```

Detección de clústeres por parte del cliente y retroceso exponencial (Valkey y Redis OSS)

Al conectarse a un clúster de ElastiCache Valkey o Redis OSS con el modo de clúster habilitado, la biblioteca cliente correspondiente debe reconocer los clústeres. Los clientes deben obtener un mapa de los slots hash de los nodos correspondientes del clúster para poder enviar las solicitudes a los nodos correctos, y evitar así la sobrecarga de rendimiento que supone gestionar las redirecciones del clúster. Como resultado, el cliente debe descubrir (o detectar) una lista completa de los slots y los nodos mapeados en dos situaciones diferentes:

- El cliente se inicializa y debe completar la configuración inicial de los slots.
- Se recibe una redirección MOVED del servidor; por ejemplo, en una conmutación por error, cuando la réplica se hace cargo de todos los slots atendidos por el nodo principal anterior, o en una nueva partición, cuando los slots se mueven del nodo principal de origen al nodo principal de destino.

Por lo general, el proceso de detección por parte del cliente se lleva a cabo enviando un comando CLUSTER SLOT o CLUSTER NODE al servidor de Valkey o Redis OSS. Recomendamos el método CLUSTER SLOT porque devuelve al cliente el conjunto de rangos de slots, así como los nodos principales y de réplica asociados. Es un método que no requiere un análisis adicional por parte del cliente y es más eficaz.

En función de la topología del clúster, el tamaño de la respuesta al comando CLUSTER SLOT puede variar según el tamaño del clúster. Los clústeres más grandes y con más nodos producen una respuesta mayor. Por lo tanto, es importante asegurarse de que la cantidad de clientes que llevan a cabo la detección de la topología del clúster no aumente de forma ilimitada. Por ejemplo, cuando la aplicación cliente se inicia o pierde la conexión con el servidor y debe realizar una detección de

clústeres, un error común es que la aplicación cliente desencadene varias solicitudes de reconexión y detección sin añadir un retroceso exponencial con el nuevo intento. Esto puede hacer que el servidor de Valkey o Redis OSS deje de responder durante mucho tiempo, con un uso de la CPU del 100 %. La interrupción se prolonga si cada comando CLUSTER SLOT debe procesar una gran cantidad de nodos en el bus del clúster. En el pasado, hemos observado varias interrupciones en los clientes debido a este comportamiento en varios lenguajes diferentes, como Python (redis-py-cluster) y Java (Lettuce y Redisson).

En una caché sin servidor, muchos de los problemas se mitigan automáticamente porque la topología de clúster expuesta es estática y consta de dos entradas: un punto de conexión de escritura y otro de lectura. La detección de clústeres también se distribuye automáticamente entre varios nodos cuando se utiliza el punto de conexión de la caché. Sin embargo, las siguientes recomendaciones siguen siendo útiles.

A fin de mitigar el impacto causado por una entrada repentina de solicitudes de conexión y detección, recomendamos lo siguiente:

- Implemente un grupo de conexiones de cliente con un tamaño finito a fin de limitar el número de conexiones entrantes simultáneas desde la aplicación cliente.
- Cuando el cliente se desconecte del servidor debido al tiempo de espera, vuelva a intentarlo con retroceso exponencial y fluctuación. Esto evitará que varios clientes sobrecarguen el servidor al mismo tiempo.
- Utilice la guía en Búsqueda de puntos de conexión en ElastiCache para encontrar el punto de conexión del clúster que necesitará a fin de realizar la detección del clúster. De este modo, distribuirá la carga de detección entre todos los nodos del clúster (hasta 90), en lugar de centrarse en unos pocos nodos raíz codificados del clúster.

A continuación, se muestran algunos ejemplos de código para la lógica de reintentos de retroceso exponencial en redis-py, PHPRedis y Lettuce.

Ejemplo 1 de lógica de retroceso: redis-py

Redis-py tiene un mecanismo de reintento incorporado: se lleva a cabo un reintento inmediatamente después de un error. Este mecanismo se puede activar mediante el argumento `retry_on_timeout` proporcionado al crear un objeto de Redis OSS. Aquí mostramos un mecanismo de reintento personalizado con retroceso exponencial y fluctuación. Hemos enviado una solicitud de extracción para implementar de forma nativa el retroceso exponencial en redis-py (1494). En el futuro, puede que no sea necesario implementarlo manualmente.

```
def run_with_backoff(function, retries=5):
base_backoff = 0.1 # base 100ms backoff
max_backoff = 10 # sleep for maximum 10 seconds
tries = 0
while True:
try:
  return function()
except (ConnectionError, TimeoutError):
  if tries >= retries:
 raise
  backoff = min(max_backoff, base_backoff * (pow(2, tries) + random.random()))
  print(f"sleeping for {backoff:.2f}s")
  sleep(backoff)
  tries += 1
```

Luego, puede utilizar el siguiente código para establecer un valor:

```
client = redis.Redis(connection_pool=redis.BlockingConnectionPool(host=HOST,
 max_connections=10))
res = run_with_backoff(lambda: client.set("key", "value"))
print(res)
```

En función de la carga de trabajo, es posible que desee cambiar el valor del retroceso base (de 1 segundo a unas pocas decenas o cientos de milisegundos) para las cargas de trabajo sensibles a la latencia.

Ejemplo 2 de lógica de retroceso: PHPRedis

PHPRedis tiene un mecanismo de reintento incorporado: hay un número máximo de reintento de 10 veces (no configurable). Hay un retraso configurable entre los intentos (con una fluctuación a partir del segundo reintento). Para obtener más información, consulte el siguiente código de muestra. Enviamos una solicitud de extracción para implementar de forma nativa el retroceso exponencial en PHPredis (1986); desde entonces, se ha fusionado y documentado. Si utiliza la última versión de PHPRedis, no será necesario implementarla manualmente, pero hemos incluido aquí la referencia para las versiones anteriores. De momento, aquí tiene un ejemplo de código para configurar el retraso del mecanismo de reintento:

```
$timeout = 0.1; // 100 millisecond connection timeout
$retry_interval = 100; // 100 millisecond retry interval
$client = new Redis();
if($client->pconnect($HOST, $PORT, $timeout, NULL, $retry_interval) != TRUE) {
```

```
 return; // ERROR: connection failed
}
$client->set($key, $value);
```

Ejemplo 3 de lógica de retroceso: Lettuce

Lettuce tiene mecanismos de reintento incorporados que emplean las estrategias de retroceso exponencial descritas en la publicación Exponential Backoff and Jitter. A continuación, puede ver un fragmento de código con el método de fluctuación total:

```
public static void main(String[] args)
{
 ClientResources resources = null;
 RedisClient client = null;

 try {
  resources = DefaultClientResources.builder()
    .reconnectDelay(Delay.fullJitter(
   Duration.ofMillis(100),     // minimum 100 millisecond delay
   Duration.ofSeconds(5),      // maximum 5 second delay
   100, TimeUnit.MILLISECONDS) // 100 millisecond base
  ).build();

  client = RedisClient.create(resources, RedisURI.create(HOST, PORT));
  client.setOptions(ClientOptions.builder()
 .socketOptions(SocketOptions.builder().connectTimeout(Duration.ofMillis(100)).build()) //
 100 millisecond connection timeout
 .timeoutOptions(TimeoutOptions.builder().fixedTimeout(Duration.ofSeconds(5)).build()) //
 5 second command timeout
 .build());

     // use the connection pool from above example
 } finally {
  if (connection != null) {
   connection.close();
  }

  if (client != null){
   client.shutdown();
  }

  if (resources != null){
   resources.shutdown();
```

```
  }

 }
}
```

Configuración de un tiempo de espera del cliente (Valkey y Redis OSS)

Configuración del tiempo de espera del cliente

Configure el tiempo de espera del lado del cliente de la manera adecuada, a fin de que el servidor tenga tiempo suficiente para procesar la solicitud y generar la respuesta. Esto también le permitirá responder rápido a los errores si no se puede establecer la conexión con el servidor. Ciertos comandos de Valkey o Redis OSS pueden ser más costosos desde el punto de vista computacional que otros. Un ejemplo de esto son los scripts de Lua o las transacciones MULTI/EXEC con varios comandos que deben ejecutarse de forma integral. Por lo general, es recomendable aumentar el tiempo de espera del cliente para evitar que este se quede sin tiempo antes de recibir la respuesta del servidor, lo que incluye lo siguiente:

- Ejecutar comandos en varias claves
- Ejecutar transacciones MULTI/EXEC o scripts de Lua que consten de varios comandos individuales de Valkey o Redis OSS
- Leer valores grandes
- Realizar operaciones de bloqueo, como BLPOP

En el caso de una operación de bloqueo como BLPOP, el procedimiento recomendado es establecer el tiempo de espera del comando en un número inferior al tiempo de espera del socket.

A continuación, encontrará ejemplos de código para implementar un tiempo de espera del lado del cliente en redis-py, PHPRedis y Lettuce.

Ejemplo 1 de configuración de tiempo de espera: redis-py

A continuación, puede ver un ejemplo de código con redis-py:

```
# connect to Redis server with a 100 millisecond timeout
# give every Redis command a 2 second timeout
client = redis.Redis(connection_pool=redis.BlockingConnectionPool(host=HOST,
 max_connections=10,socket_connect_timeout=0.1,socket_timeout=2))

res = client.set("key", "value") # will timeout after 2 seconds
```

```
print(res)                       # if there is a connection error

res = client.blpop("list", timeout=1) # will timeout after 1 second
                                      # less than the 2 second socket timeout
print(res)
```

Ejemplo 2 de configuración de tiempo de espera: PHPRedis

A continuación, puede ver un ejemplo de código con PHPRedis:

```
// connect to Redis server with a 100ms timeout
// give every Redis command a 2s timeout
$client = new Redis();
$timeout = 0.1; // 100 millisecond connection timeout
$retry_interval = 100; // 100 millisecond retry interval
$client = new Redis();
if($client->pconnect($HOST, $PORT, 0.1, NULL, 100, $read_timeout=2) != TRUE){
 return; // ERROR: connection failed
}
$client->set($key, $value);

$res = $client->set("key", "value"); // will timeout after 2 seconds
print "$res\n";                      // if there is a connection error

$res = $client->blpop("list", 1); // will timeout after 1 second
print "$res\n";                   // less than the 2 second socket timeout
```

Ejemplo 3 de configuración de tiempo de espera: Lettuce

A continuación, puede ver un ejemplo de código con Lettuce:

```
// connect to Redis server and give every command a 2 second timeout
public static void main(String[] args)
{
 RedisClient client = null;
 StatefulRedisConnection<String, String> connection = null;
 try {
  client = RedisClient.create(RedisURI.create(HOST, PORT));
  client.setOptions(ClientOptions.builder()
 .socketOptions(SocketOptions.builder().connectTimeout(Duration.ofMillis(100)).build()) //
 100 millisecond connection timeout
 .timeoutOptions(TimeoutOptions.builder().fixedTimeout(Duration.ofSeconds(2)).build()) //
 2 second command timeout
```

```
  .build());

   // use the connection pool from above example

   commands.set("key", "value"); // will timeout after 2 seconds
   commands.blpop(1, "list"); // BLPOP with 1 second timeout
  } finally {
   if (connection != null) {
    connection.close();
   }

   if (client != null){
    client.shutdown();
   }
  }
 }
```

Configuración de un tiempo de espera del servidor (Valkey y Redis OSS)

Hemos observado casos en los que la aplicación de un cliente tiene un gran número de clientes inactivos conectados, pero no envía comandos de forma activa. En estos casos, se pueden agotar las 65 000 conexiones con un número elevado de clientes inactivos. Para evitar este tipo de situaciones, configure el tiempo de espera adecuadamente en el servidor mediante Parámetros de Valkey y Redis OSS. Esto garantiza que el servidor tome la iniciativa en desconectar a los clientes inactivos para evitar un aumento en el número de conexiones. Esta configuración no está disponible en las caché sin servidor.

Scripts de Lua

Valkey y Redis OSS admiten más de 200 comandos, incluidos los que ejecutan scripts de Lua. Sin embargo, en lo que respecta a los scripts de Lua, hay varios inconvenientes que pueden afectar a la memoria y a la disponibilidad de Valkey o Redis OSS.

Scripts de Lua no parametrizados

Cada script de Lua se almacena en caché en el servidor de Valkey o Redis OSS antes de ejecutarse. Los scripts de Lua no parametrizados son únicos, lo que puede provocar que el servidor de Valkey o Redis OSS almacene una gran cantidad de ellos y consuma más memoria. Para mitigar esta situación, asegúrese de que todos los scripts de Lua estén parametrizados y ejecute SCRIPT FLUSH con regularidad a fin de depurar los scripts de Lua almacenados en caché si es necesario.

También tenga en cuenta que se deben proporcionar claves. Si no se proporciona ningún valor para el parámetro KEY, el script fallará. Por ejemplo, lo siguiente no funcionará:

```
serverless-test-lst4hg.serverless.use1.cache.amazonaws.com:6379> eval 'return "Hello
 World"' 0
(error) ERR Lua scripts without any input keys are not supported.
```

Esto sí funcionará:

```
serverless-test-lst4hg.serverless.use1.cache.amazonaws.com:6379> eval 'return
 redis.call("get", KEYS[1])' 1 mykey-2
"myvalue-2"
```

En el siguiente ejemplo, se muestra cómo definir y utilizar parámetros: En primer lugar, tenemos el ejemplo de un método no parametrizado que da como resultado tres scripts de Lua en caché diferentes (no recomendable):

```
eval "return redis.call('set','key1','1')" 0
eval "return redis.call('set','key2','2')" 0
eval "return redis.call('set','key3','3')" 0
```

En lugar de esto, utilice el siguiente patrón para crear un único script que pueda aceptar los parámetros pasados:

```
eval "return redis.call('set',KEYS[1],ARGV[1])" 1 key1 1
eval "return redis.call('set',KEYS[1],ARGV[1])" 1 key2 2
eval "return redis.call('set',KEYS[1],ARGV[1])" 1 key3 3
```

Scripts de Lua de larga duración

Los scripts de Lua pueden ejecutar varios comandos de forma atómica, por lo que pueden tardar más en completarse que un comando normal de Valkey o Redis OSS. Si el script de Lua solo ejecuta operaciones de solo lectura, puede detenerlo durante la ejecución. Sin embargo, cuando el script de Lua lleve a cabo operaciones de escritura, no se puede detener y debe ejecutarse hasta el final. Un script de Lua de ejecución prolongada que esté mutando puede provocar que el servidor de Valkey o Redis OSS deje de responder durante mucho tiempo. Para mitigar este problema, evite el uso de scripts de Lua de larga duración y pruébelos en un entorno de preproducción.

Script de Lua con escrituras encubiertas

Aunque Valkey o Redis OSS hayan superado la `maxmemory`, un script de Lua puede seguir escribiendo nuevos datos en Valkey o Redis OSS de varias maneras:

- El script se inicia cuando el servidor de Valkey o Redis OSS está por debajo de `maxmemory` y contiene varias operaciones de escritura en su interior.
- El primer comando de escritura del script no consume memoria (como DEL) y va seguido de más operaciones de escritura que sí consumen memoria.
- Puede mitigar este problema configurando una política de expulsión adecuada en un servidor de Valkey o Redis OSS que no sea `noeviction`. Esto permite que Redis OSS expulse elementos y libere memoria entre los scripts de Lua.

Almacenamiento de elementos compuestos de gran tamaño (Valkey y Redis OSS)

En algunos casos, una aplicación puede almacenar elementos compuestos de gran tamaño en Valkey o Redis OSS (como un conjunto de datos hash de varios GB). Esto no es recomendable, ya que suele provocar problemas de rendimiento en Valkey o Redis OSS. Por ejemplo, el cliente puede ejecutar un comando HGETALL para recuperar toda la colección de hash de varios GB. Se puede generar una presión en la memoria importante para el servidor de Valkey o Redis OSS que almacena el elemento de gran tamaño en el búfer de salida del cliente. Además, para la migración de slots en modo clúster, ElastiCache no migra los slots que contienen elementos con un tamaño serializado superior a 256 MB.

Para resolver los problemas implicados en la gestión de elementos de gran tamaño, recomendamos lo siguiente:

- Divida el elemento compuesto grande en varios elementos más pequeños. Algo que puede hacer es dividir una colección hash grande en campos clave-valor individuales con un esquema de nombres de clave que refleje adecuadamente la colección; por ejemplo, puede utilizar un prefijo común en el nombre de la clave para identificar la colección de elementos. Si debe acceder a varios campos de la misma colección de forma integral, puede usar el comando MGET para recuperar varios valores clave en el mismo comando.
- Si ha evaluado todas las opciones y sigue sin poder dividir el conjunto de datos de gran tamaño de la colección, intente utilizar comandos que funcionen en un subconjunto de los datos de la colección, en lugar de en toda la colección. Evite los casos de uso que requieran recuperar de forma integral toda la colección de varios GB con el mismo comando. Un ejemplo de esto es el uso de los comandos HGET o HMGET, en lugar de HGETALL, en las colecciones de hash.

Configuración del cliente de Lettuce (Valkey y Redis OSS)

En esta sección se describen las opciones de configuración recomendadas de Java y Lettuce y cómo se aplican a los clústeres de ElastiCache.

Las recomendaciones de esta sección se probaron con la versión 6.2.2 de Lettuce.

Temas

- Ejemplo: Configuración de Lettuce para el modo de clúster con TLS habilitado
- Ejemplo: Configuración de Lettuce para el modo de clúster deshabilitado con TLS habilitado

TTL de la caché DNS de Java

La máquina virtual de Java (JVM) almacena en caché las búsquedas de nombres DNS. Cuando la JVM resuelve un nombre de host en una dirección IP, almacena en caché la dirección IP durante un periodo de tiempo especificado, conocido como tiempo de vida (TTL).

La elección del valor de TTL implica un compromiso entre latencia y la capacidad de respuesta a los cambios. Con TTL más cortos, los solucionadores de DNS detectan las actualizaciones en el DNS del clúster con mayor rapidez. Esto puede hacer que la aplicación responda más rápido a las sustituciones u otros flujos de trabajo a los que se somete el clúster. Sin embargo, si el TTL es demasiado bajo, aumenta el volumen de consultas, lo que puede aumentar la latencia de la aplicación. Aunque no existe un valor de TTL correcto, vale la pena esperar a que surta efecto un cambio cuando se configura el valor de TTL.

Como los nodos de ElastiCache utilizan entradas de nombres de DNS que pueden cambiar de vez en cuando, recomendamos que configure JVM con un TTL bajo de 5 a 10 segundos. Con esto, se asegurará de que, cuando cambie la dirección IP de un nodo, su aplicación pueda recibir y utilizar la nueva dirección IP del recurso volviendo a consultar la entrada de DNS.

En algunas configuraciones de Java, el TTL predeterminado de JVM está establecido de forma que nunca se actualicen las entradas DNS hasta que se reinicie la JVM.

Para obtener más información sobre cómo configurar el TTL de JVM, consulte Cómo configurar el TTL de JVM.

Versión de Lettuce

Recomendamos usar la versión 6.2.2 o posterior de Lettuce.

puntos de conexión

Cuando utilice clústeres habilitados para el modo de clúster, establezca `redisUri` en el punto de conexión de configuración del clúster. La búsqueda de DNS para este URI devuelve una lista de todos los nodos disponibles en el clúster y se resuelve aleatoriamente en uno de ellos durante la inicialización del clúster. Para obtener más información sobre cómo funciona la actualización de topología, consulte dynamicRefreshResources más adelante en este tema.

SocketOption

Habilite KeepAlive. Al habilitar esta opción, se reduce la necesidad de gestionar las conexiones erróneas durante el tiempo de ejecución del comando.

Asegúrese de configurar el Tiempo de espera de la conexión en función de los requisitos de la aplicación y la carga de trabajo. Para obtener más información, consulte la sección de tiempos de espera más adelante en este tema.

ClusterClientOption: Opciones de cliente habilitadas para el modo de clúster

Habilite AutoReconnect cuando se pierda la conexión.

Establezca CommandTimeout. Para obtener más información, consulte la sección Tiempos de espera más adelante en este tema.

Establezca nodeFilter para filtrar los nodos con errores de la topología. Lettuce guarda todos los nodos que se encuentran en la salida de los “nodos del clúster” (incluidos los nodos con el estado PFAIL/FAIL) en las “particiones” del cliente. Durante el proceso de creación de la topología del clúster, intenta conectarse a todos los nodos de partición. Este comportamiento de Lettuce de agregar nodos con errores puede provocar errores de conexión (o advertencias) cuando los nodos se sustituyen por cualquier motivo.

Por ejemplo, una vez que finaliza una conmutación por error y el clúster inicia el proceso de recuperación, mientras se actualiza la topología del clúster, el mapa de nodos del bus del clúster tiene un breve periodo de tiempo en el que el nodo inactivo se muestra como nodo FAIL, antes de que se elimine por completo de la topología. Durante este periodo, el cliente de Lettuce lo considera un nodo en buen estado y se conecta continuamente a él. Esto provoca un error cuando se agota el reintento.

Por ejemplo:

```
final ClusterClientOptions clusterClientOptions =
    ClusterClientOptions.builder()
```

```
    ... // other options
    .nodeFilter(it ->
        ! (it.is(RedisClusterNode.NodeFlag.FAIL)
        || it.is(RedisClusterNode.NodeFlag.EVENTUAL_FAIL)
        || it.is(RedisClusterNode.NodeFlag.HANDSHAKE)
        || it.is(RedisClusterNode.NodeFlag.NOADDR)))
    .validateClusterNodeMembership(false)
    .build();
redisClusterClient.setOptions(clusterClientOptions);
```

> **Note**
>
> El filtrado de nodos se utiliza mejor con DynamicRefreshSources configurado en verdadero. De lo contrario, si la vista de topología se toma de un solo nodo raíz problemático, que ve que un nodo principal de alguna partición está produciendo un error, filtrará este nodo principal, lo que hará que las ranuras no queden cubiertas. Tener varios nodos raíz (cuando DynamicRefreshSources es verdadero) reduce la probabilidad de que se produzca este problema, ya que al menos algunos de los nodos raíz deberían tener una vista de topología actualizada después de una conmutación por error con el nodo principal recién promocionado.

ClusterTopologyRefreshOptions: opciones para controlar la actualización de la topología del clúster del cliente habilitado para el modo de clúster

> **Note**
>
> Los clústeres desactivados en modo de clúster no admiten los comandos de detección de clústeres y no son compatibles con la funcionalidad de detección de topología dinámica de todos los clientes.
>
> El modo de clúster desactivado con ElastiCache no es compatible con `MasterSlaveTopologyRefresh` de Lettuce. En cambio, para el modo de clúster desactivado puede configurar `StaticMasterReplicaTopologyProvider` y proporcionar los puntos de conexión de lectura y escritura del clúster.
>
> Para obtener más información acerca de la conexión a clústeres desactivados en modo de clúster, consulte Búsqueda de puntos de conexión de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) (consola).
>
> Si desea utilizar la funcionalidad de detección de topología dinámica de Lettuce, puede crear un clúster habilitado para el modo de clúster con la configuración del mismo fragmento que

el clúster existente. Sin embargo, para los clústeres habilitados para el modo de clúster, recomendamos configurar al menos 3 particiones con al menos 1 réplica para admitir una conmutación por error rápida.

Habilite enablePeriodicRefresh. Esto permite las actualizaciones periódicas de la topología del clúster para que el cliente actualice la topología del clúster en los intervalos del periodo de actualización (predeterminado: 60 segundos). Cuando están desactivadas, el cliente actualiza la topología del clúster solo cuando se producen errores al intentar ejecutar comandos en el clúster.

Con esta opción habilitada, puede reducir la latencia asociada a la actualización de la topología del clúster agregando este trabajo a una tarea que se esté ejecutando en segundo plano. Aunque la actualización de la topología se realiza en un trabajo que se esté ejecutando en segundo plano, puede resultar algo lenta para los clústeres con muchos nodos. Esto se debe a que se están consultando las vistas de todos los nodos para obtener la vista de clúster más actualizada. Si ejecuta un clúster grande, es posible que desee aumentar el periodo.

Habilite enableAllAdaptiveRefreshTriggers. Esto permite actualizar la topología adaptativa y utilizar todos los disparadores: MOVED_REDIRECT, ASK_REDIRECT, PERSISTENT_RECONNECTS, UNCOVERED_SLOT, UNKNOWN_NODE. Los desencadenadores de actualización adaptativa inician las actualizaciones de la vista de topología en función de los eventos que se producen durante las operaciones de clúster de Valkey y Redis OSS. Al habilitar esta opción, se actualiza inmediatamente la topología cuando se activa uno de los desencadenadores anteriores. Las actualizaciones desencadenadas adaptativas tienen un límite de velocidad mediante un tiempo de espera porque los eventos se pueden producir a gran escala (tiempo de espera predeterminado entre actualizaciones: 30).

Habilite closeStaleConnections. Esto permite cerrar las conexiones obsoletas al actualizar la topología del clúster. Solo entra en vigor si ClusterTopologyRefreshOptions.isPeriodicRefreshEnabled() es verdadero. Cuando se habilita, el cliente puede cerrar las conexiones obsoletas y crear otras nuevas en segundo plano. Esto reduce la necesidad de gestionar las conexiones erróneas durante el tiempo de ejecución del comando.

Habilite dynamicRefreshResources. Recomendamos habilitar dynamicRefreshResources para clústeres pequeños y desactivarlo para clústeres grandes. dynamicRefreshResources permite detectar los nodos del clúster desde el nodo raíz proporcionado (por ejemplo, el punto de conexión de configuración del clúster). Utiliza todos los nodos detectados como orígenes para actualizar la topología del clúster.

El uso de la actualización dinámica consulta todos los nodos detectados para la topología del clúster e intenta elegir la vista de clúster más precisa. Si se establece en falso, solo se utilizan los nodos raíz iniciales como orígenes para la detección de la topología y el número de clientes se obtiene solo para los nodos raíz iniciales. Cuando se desactiva, si el punto de conexión de la configuración del clúster se resuelve como un nodo con errores, se produce un error al intentar actualizar la vista del clúster y se producen excepciones. Este escenario se puede producir porque pasa algún tiempo hasta que la entrada de un nodo con errores se elimina del punto de conexión de la configuración del clúster. Por lo tanto, el punto de conexión de la configuración aún se puede resolver aleatoriamente en un nodo erróneo durante un breve periodo de tiempo.

Sin embargo, cuando se habilita, utilizamos todos los nodos del clúster que se reciben de la vista de clústeres para consultar la vista actual. Como filtramos los nodos con errores de esa vista, la actualización de la topología se realizará correctamente. Sin embargo, cuando dynamicRefreshSources es verdadero, Lettuce consulta todos los nodos para obtener la vista del clúster y, a continuación, compara los resultados. Por lo tanto, puede resultar caro para los clústeres con muchos nodos. Le sugerimos que desactive esta característica para los clústeres con muchos nodos.

```
final ClusterTopologyRefreshOptions topologyOptions =
    ClusterTopologyRefreshOptions.builder()
    .enableAllAdaptiveRefreshTriggers()
    .enablePeriodicRefresh()
    .dynamicRefreshSources(true)
    .build();
```

ClientResources

Configure DnsResolver con DirContextDnsResolver. El servicio de resolución de DNS se basa en com.sun.jndi.dns.DnsContextFactory de Java.

Configure reconnectDelay con retroceso exponencial y fluctuación total. Lettuce tiene mecanismos de reintento integrados basados en las estrategias de retroceso exponencial. Para obtener más información, consulte Retroceso exponencial y fluctuación en el blog de arquitectura de AWS. Para obtener más información sobre la importancia de contar con una estrategia de reintento, consulte las secciones sobre lógica de retraso de la Publicación del blog sobre mejores prácticas en el blog de base de datos de AWS.

```
ClientResources clientResources = DefaultClientResources.builder()
   .dnsResolver(new DirContextDnsResolver())
    .reconnectDelay(
```

```
        Delay.fullJitter(
            Duration.ofMillis(100),     // minimum 100 millisecond delay
            Duration.ofSeconds(10),      // maximum 10 second delay
            100, TimeUnit.MILLISECONDS)) // 100 millisecond base
    .build();
```

Tiempos de espera

Utilice un valor de tiempo de espera de conexión inferior al tiempo de espera del comando. Lettuce utiliza un establecimiento de conexión diferida. Por lo tanto, si el tiempo de espera de conexión es superior al tiempo de espera del comando, puede producirse un periodo de error persistente tras una actualización de la topología si Lettuce intenta conectarse a un nodo en mal estado y siempre se supera el tiempo de espera del comando.

Utilice un tiempo de espera de comando dinámico para diferentes comandos. Le recomendamos que establezca el tiempo de espera del comando en función de la duración esperada del comando. Por ejemplo, utilice un tiempo de espera más largo para los comandos que se repiten en varias claves, como los scripts FLUSHDB, FLUSHALL, KEYS, SMEMBERS o Lua. Utilice tiempos de espera más cortos para los comandos de una sola clave, como SET, GET y HSET.

> **Note**
>
> Los tiempos de espera que se configuran en el siguiente ejemplo son para pruebas que ejecutaron comandos SET/GET con claves y valores de hasta 20 bytes de longitud. El tiempo de procesamiento puede ser mayor cuando los comandos son complejos o las claves y los valores son más grandes. Debe establecer los tiempos de espera en función del caso de uso de la aplicación.

```
private static final Duration META_COMMAND_TIMEOUT = Duration.ofMillis(1000);
private static final Duration DEFAULT_COMMAND_TIMEOUT = Duration.ofMillis(250);
// Socket connect timeout should be lower than command timeout for Lettuce
private static final Duration CONNECT_TIMEOUT = Duration.ofMillis(100);

SocketOptions socketOptions = SocketOptions.builder()
    .connectTimeout(CONNECT_TIMEOUT)
    .build();


class DynamicClusterTimeout extends TimeoutSource {
```

```
     private static final Set<ProtocolKeyword> META_COMMAND_TYPES =
 ImmutableSet.<ProtocolKeyword>builder()
          .add(CommandType.FLUSHDB)
          .add(CommandType.FLUSHALL)
          .add(CommandType.CLUSTER)
          .add(CommandType.INFO)
          .add(CommandType.KEYS)
          .build();

    private final Duration defaultCommandTimeout;
    private final Duration metaCommandTimeout;

    DynamicClusterTimeout(Duration defaultTimeout, Duration metaTimeout)
    {
        defaultCommandTimeout = defaultTimeout;
        metaCommandTimeout = metaTimeout;
    }

    @Override
    public long getTimeout(RedisCommand<?, ?, ?> command) {
        if (META_COMMAND_TYPES.contains(command.getType())) {
            return metaCommandTimeout.toMillis();
        }
        return defaultCommandTimeout.toMillis();
    }
}

// Use a dynamic timeout for commands, to avoid timeouts during
// cluster management and slow operations.
TimeoutOptions timeoutOptions = TimeoutOptions.builder()
.timeoutSource(
    new DynamicClusterTimeout(DEFAULT_COMMAND_TIMEOUT, META_COMMAND_TIMEOUT))
.build();
```

Ejemplo: Configuración de Lettuce para el modo de clúster con TLS habilitado

> **Note**
>
> Los tiempos de espera del siguiente ejemplo son para pruebas que ejecutaron comandos SET/GET con claves y valores de hasta 20 bytes de longitud. El tiempo de procesamiento puede ser mayor cuando los comandos son complejos o las claves y los valores son más grandes. Debe establecer los tiempos de espera en función del caso de uso de la aplicación.

```
// Set DNS cache TTL
public void setJVMProperties() {
    java.security.Security.setProperty("networkaddress.cache.ttl", "10");
}

private static final Duration META_COMMAND_TIMEOUT = Duration.ofMillis(1000);
private static final Duration DEFAULT_COMMAND_TIMEOUT = Duration.ofMillis(250);
// Socket connect timeout should be lower than command timeout for Lettuce
private static final Duration CONNECT_TIMEOUT = Duration.ofMillis(100);

// Create RedisURI from the cluster configuration endpoint
clusterConfigurationEndpoint = <cluster-configuration-endpoint> // TODO: add your
 cluster configuration endpoint
final RedisURI redisUriCluster =
    RedisURI.Builder.redis(clusterConfigurationEndpoint)
        .withPort(6379)
        .withSsl(true)
        .build();

// Configure the client's resources
ClientResources clientResources = DefaultClientResources.builder()
    .reconnectDelay(
        Delay.fullJitter(
            Duration.ofMillis(100),     // minimum 100 millisecond delay
            Duration.ofSeconds(10),      // maximum 10 second delay
            100, TimeUnit.MILLISECONDS)) // 100 millisecond base
    .dnsResolver(new DirContextDnsResolver())
    .build();

// Create a cluster client instance with the URI and resources
RedisClusterClient redisClusterClient =
    RedisClusterClient.create(clientResources, redisUriCluster);

// Use a dynamic timeout for commands, to avoid timeouts during
// cluster management and slow operations.
class DynamicClusterTimeout extends TimeoutSource {
     private static final Set<ProtocolKeyword> META_COMMAND_TYPES =
 ImmutableSet.<ProtocolKeyword>builder()
          .add(CommandType.FLUSHDB)
          .add(CommandType.FLUSHALL)
          .add(CommandType.CLUSTER)
          .add(CommandType.INFO)
          .add(CommandType.KEYS)
```

```
            .build();

    private final Duration metaCommandTimeout;
    private final Duration defaultCommandTimeout;

    DynamicClusterTimeout(Duration defaultTimeout, Duration metaTimeout)
    {
        defaultCommandTimeout = defaultTimeout;
        metaCommandTimeout = metaTimeout;
    }

    @Override
    public long getTimeout(RedisCommand<?, ?, ?> command) {
        if (META_COMMAND_TYPES.contains(command.getType())) {
            return metaCommandTimeout.toMillis();
        }
        return defaultCommandTimeout.toMillis();
    }
}

TimeoutOptions timeoutOptions = TimeoutOptions.builder()
    .timeoutSource(new DynamicClusterTimeout(DEFAULT_COMMAND_TIMEOUT,
 META_COMMAND_TIMEOUT))
     .build();

// Configure the topology refreshment options
final ClusterTopologyRefreshOptions topologyOptions =
    ClusterTopologyRefreshOptions.builder()
    .enableAllAdaptiveRefreshTriggers()
    .enablePeriodicRefresh()
    .dynamicRefreshSources(true)
    .build();

// Configure the socket options
final SocketOptions socketOptions =
    SocketOptions.builder()
    .connectTimeout(CONNECT_TIMEOUT)
    .keepAlive(true)
    .build();

// Configure the client's options
final ClusterClientOptions clusterClientOptions =
    ClusterClientOptions.builder()
    .topologyRefreshOptions(topologyOptions)
```

```
    .socketOptions(socketOptions)
    .autoReconnect(true)
    .timeoutOptions(timeoutOptions)
    .nodeFilter(it ->
        ! (it.is(RedisClusterNode.NodeFlag.FAIL)
        || it.is(RedisClusterNode.NodeFlag.EVENTUAL_FAIL)
        || it.is(RedisClusterNode.NodeFlag.NOADDR)))
    .validateClusterNodeMembership(false)
    .build();

redisClusterClient.setOptions(clusterClientOptions);

// Get a connection
final StatefulRedisClusterConnection<String, String> connection =
    redisClusterClient.connect();

// Get cluster sync/async commands
RedisAdvancedClusterCommands<String, String> sync = connection.sync();
RedisAdvancedClusterAsyncCommands<String, String> async = connection.async();
```

Ejemplo: Configuración de Lettuce para el modo de clúster deshabilitado con TLS habilitado

> **Note**
>
> Los tiempos de espera del siguiente ejemplo son para pruebas que ejecutaron comandos SET/GET con claves y valores de hasta 20 bytes de longitud. El tiempo de procesamiento puede ser mayor cuando los comandos son complejos o las claves y los valores son más grandes. Debe establecer los tiempos de espera en función del caso de uso de la aplicación.

```
// Set DNS cache TTL
public void setJVMProperties() {
    java.security.Security.setProperty("networkaddress.cache.ttl", "10");
}

private static final Duration META_COMMAND_TIMEOUT = Duration.ofMillis(1000);
private static final Duration DEFAULT_COMMAND_TIMEOUT = Duration.ofMillis(250);
// Socket connect timeout should be lower than command timeout for Lettuce
private static final Duration CONNECT_TIMEOUT = Duration.ofMillis(100);

// Create RedisURI from the primary/reader endpoint
clusterEndpoint = <primary/reader-endpoint> // TODO: add your node endpoint
```

```
RedisURI redisUriStandalone =

 RedisURI.Builder.redis(clusterEndpoint).withPort(6379).withSsl(true).withDatabase(0).build();

ClientResources clientResources =
    DefaultClientResources.builder()
        .dnsResolver(new DirContextDnsResolver())
        .reconnectDelay(
            Delay.fullJitter(
                Duration.ofMillis(100), // minimum 100 millisecond delay
                Duration.ofSeconds(10), // maximum 10 second delay
                100,
                TimeUnit.MILLISECONDS)) // 100 millisecond base
        .build();

// Use a dynamic timeout for commands, to avoid timeouts during
// slow operations.
class DynamicTimeout extends TimeoutSource {
     private static final Set<ProtocolKeyword> META_COMMAND_TYPES =
 ImmutableSet.<ProtocolKeyword>builder()
          .add(CommandType.FLUSHDB)
          .add(CommandType.FLUSHALL)
          .add(CommandType.INFO)
          .add(CommandType.KEYS)
          .build();

    private final Duration metaCommandTimeout;
    private final Duration defaultCommandTimeout;

    DynamicTimeout(Duration defaultTimeout, Duration metaTimeout)
    {
        defaultCommandTimeout = defaultTimeout;
        metaCommandTimeout = metaTimeout;
    }

    @Override
    public long getTimeout(RedisCommand<?, ?, ?> command) {
        if (META_COMMAND_TYPES.contains(command.getType())) {
            return metaCommandTimeout.toMillis();
        }
        return defaultCommandTimeout.toMillis();
    }
}
```

```
TimeoutOptions timeoutOptions = TimeoutOptions.builder()
    .timeoutSource(new DynamicTimeout(DEFAULT_COMMAND_TIMEOUT, META_COMMAND_TIMEOUT))
     .build();

final SocketOptions socketOptions =
    SocketOptions.builder().connectTimeout(CONNECT_TIMEOUT).keepAlive(true).build();

ClientOptions clientOptions =

 ClientOptions.builder().timeoutOptions(timeoutOptions).socketOptions(socketOptions).build();

RedisClient redisClient = RedisClient.create(clientResources, redisUriStandalone);
redisClient.setOptions(clientOptions);
```

Configuración de un protocolo preferido para clústeres de doble pila (Valkey y Redis OSS)

En el caso de los clústeres de Valkey o Redis OSS habilitados para el modo de clúster, puede controlar el protocolo que los clientes utilizarán para conectarse a los nodos del clúster con el parámetro de detección de IP. El parámetro de detección de IP se puede establecer en IPv4 o IPv6.

Para los clústeres de Valkey o Redis OSS, el parámetro de detección de IP establece el protocolo IP utilizado en la salida de las ranuras del clúster (), las particiones del clúster () y los nodos del clúster (). Los clientes utilizan estos comandos para detectar la topología del clúster. Los clientes usan las IP de estos comandos para conectarse a los otros nodos del clúster.

Cambiar la detección de IP no provocará ningún tiempo de inactividad para los clientes conectados. Sin embargo, los cambios tardarán algún tiempo en propagarse. Para determinar cuándo los cambios se han propagado por completo para un clúster de Valkey o Redis OSS, supervise la salida de `cluster slots`. Una vez que todos los nodos devueltos por el comando de ranuras del clúster registren las IP con el nuevo protocolo, los cambios terminarán de propagarse.

Ejemplo con Redis-Py:

```
cluster = RedisCluster(host="xxxx", port=6379)
target_type = IPv6Address # Or IPv4Address if changing to IPv4

nodes = set()
while len(nodes) == 0 or not all((type(ip_address(host)) is target_type) for host in
 nodes):
    nodes = set()
```

```
   # This refreshes the cluster topology and will discovery any node updates.
   # Under the hood it calls cluster slots
    cluster.nodes_manager.initialize()
    for node in cluster.get_nodes():
        nodes.add(node.host)
    self.logger.info(nodes)

    time.sleep(1)
```

Ejemplo con Lettuce:

```
RedisClusterClient clusterClient = RedisClusterClient.create(RedisURI.create("xxxx",
 6379));

Class targetProtocolType = Inet6Address.class; // Or Inet4Address.class if you're
 switching to IPv4

Set<String> nodes;

do {
   // Check for any changes in the cluster topology.
   // Under the hood this calls cluster slots
    clusterClient.refreshPartitions();
    Set<String> nodes = new HashSet<>();

    for (RedisClusterNode node : clusterClient.getPartitions().getPartitions()) {
        nodes.add(node.getUri().getHost());
    }

    Thread.sleep(1000);
} while (!nodes.stream().allMatch(node -> {
            try {
                return finalTargetProtocolType.isInstance(InetAddress.getByName(node));
            } catch (UnknownHostException ignored) {}
            return false;
}));
```

Prácticas recomendadas para clientes (Memcached)

Conozca las prácticas recomendadas para escenarios comunes con clústeres de ElastiCache para Memcached.

Temas

- Configuración de su cliente de ElastiCache para un equilibrio de carga eficaz (Memcached)
- Clientes validados con Memcached
- Configuración de un protocolo preferido para clústeres de doble pila (Memcached)

Configuración de su cliente de ElastiCache para un equilibrio de carga eficaz (Memcached)

> **Note**
>
> Esta sección se aplica a clústeres de Memcached de varios nodos basados en nodos.

Para utilizar varios nodos de Memcached de ElastiCache de forma eficaz, debe poder repartir sus claves de caché entre los nodos. Una manera sencilla de equilibrar la carga de un clúster con n nodos es calcular el hash de la clave del objeto y aplicar la función mod al resultado mediante n: `hash(key) mod n`. El valor resultante (de 0 a n-1) es el número del nodo en el que deberá colocar el objeto.

Este enfoque es sencillo y funciona bien siempre que el número de nodos (n) sea constante. Sin embargo, siempre que agregue o elimine un nodo del clúster, el número de claves que deben moverse será (n (1)/n (donde n es el nuevo número de nodos). Por lo tanto, este enfoque da como resultado el traslado de un gran número de claves, lo que se traduce en un gran número de pérdidas iniciales de caché, especialmente cuando el número de nodos es elevado. En el mejor de los casos, al escalar de 1 a 2 resultados de nodos, se obtienen (2-1)/2 (50 %) de claves para trasladar. Al escalar de 9 a 10 nodos, se obtienen (10-1)/10 (90 %) de claves para trasladar. Si va a ampliar debido a un pico de tráfico, no deseará tener muchas pérdidas de caché. Un gran número de pérdidas de caché devuelve coincidencias con la base de datos, que ya está sobrecargada por el pico de tráfico.

La solución a este dilema es un uso consistente de la función hash. Una utilización consistente de hash emplea un algoritmo según el cual, siempre que se agregue o elimine un nodo de un clúster, el número de claves que deba moverse será aproximadamente 1/n (donde n es el número de nodos nuevo). En el peor de los casos, al escalar de 1 a 2 resultados de nodos, se obtienen 1/2 (50 por ciento) de claves para trasladar. Al escalar de 9 a 10 nodos, se obtienen 1/10 (10 por ciento) de claves para trasladar.

Como usuario, deberá controlar qué algoritmo de hash se usa para los clústeres de varios nodos. Recomendamos configurar sus clientes para que utilicen hash de forma consistente. Afortunadamente, hay muchas bibliotecas de cliente de Memcached en la mayoría de los idiomas comunes que implementan hash de forma consistente. Consulte la documentación de la biblioteca que va a utilizar para ver si admite el uso consistente de hash y saber cómo implementarlo.

Si trabaja en Java, PHP o .NET, recomendamos utilizar una de las bibliotecas de cliente de Amazon ElastiCache.

Uso consistente de hash con Java

El cliente de Java de Memcached de ElastiCache se basa en el cliente de Java spymemcached de código abierto, que tiene capacidades de utilización consistente de hash integradas. La biblioteca incluye una clase KetamaConnectionFactory que implementa hash de forma consistente. De forma predeterminada, el uso consistente de hash está desactivado en spymemcached.

Para obtener más información, consulte la documentación de KetamaConnectionFactory en KetamaConnectionFactory.

Uso consistente de hash utilizando PHP con Memcached

El cliente de PHP de Memcached de ElastiCache es un encapsulador en torno a la biblioteca integrada de PHP de Memcached. De forma predeterminada, el uso consistente de hash está desactivado en la biblioteca PHP de Memcached.

Utilice el siguiente código para habilitar el uso consistente de hash.

```
$m = new Memcached();
$m->setOption(Memcached::OPT_DISTRIBUTION, Memcached::DISTRIBUTION_CONSISTENT);
```

Además del código anterior, recomendamos habilitar también `memcached.sess_consistent_hash` en su archivo php.ini.

Para obtener más información, consulte la documentación de configuración de tiempo de ejecución para PHP de Memcached en http://php.net/manual/en/memcached.configuration.php. Tenga en cuenta específicamente el parámetro `memcached.sess_consistent_hash`.

Uso consistente de hash utilizando .NET con Memcached

El cliente de .NET de Memcached de ElastiCache es un encapsulador en torno a Enyim Memcached. De forma predeterminada, el uso consistente de hash está habilitado en el cliente Enyim Memcached.

Para obtener más información, consulte la documentación de `memcached/locator` en https://github.com/enyim/EnyimMemcached/wiki/MemcachedClient-Configuration#user-content-memcachedlocator.

Clientes validados con Memcached

Los siguientes clientes se han validado específicamente para funcionar con todas las configuraciones de tipo de red admitidas para Memcached.

Clientes validados:

- AWS ElastiCache Cluster Client Memcached para Php: versión *3.6.2
- AWS ElastiCache Cluster Client Memcached para Java: versión principal más reciente en Github

Configuración de un protocolo preferido para clústeres de doble pila (Memcached)

Para los clústeres de Memcached, puede controlar el protocolo que los clientes utilizarán para conectarse a los nodos del clúster con el parámetro de detección de IP. El parámetro de detección de IP se puede establecer en IPv4 o IPv6.

El parámetro de detección de IP controla el protocolo IP utilizado en la salida del clúster config get. Lo que, a su vez, determinará el protocolo IP utilizado por los clientes que admiten la detección automática de clústeres de ElastiCache para Memcached.

Cambiar la detección de IP no provocará ningún tiempo de inactividad para los clientes conectados. Sin embargo, los cambios tardarán algún tiempo en propagarse.

Monitoree la salida de `getAvailableNodeEndPoints` para Java y para que Php monitoree la salida de `getServerList`. Una vez que la salida de estas funciones registre las IP de todos los nodos del clúster que utilizan el protocolo actualizado, los cambios terminarán de propagarse.

Ejemplo de Java:

```
MemcachedClient client = new MemcachedClient(new InetSocketAddress("xxxx", 11211));

Class targetProtocolType = Inet6Address.class; // Or Inet4Address.class if you're
 switching to IPv4

Set<String> nodes;
```

```
do {
    nodes =
 client.getAvailableNodeEndPoints().stream().map(NodeEndPoint::getIpAddress).collect(Collectors

    Thread.sleep(1000);
} while (!nodes.stream().allMatch(node -> {
            try {
                return finalTargetProtocolType.isInstance(InetAddress.getByName(node));
            } catch (UnknownHostException ignored) {}
            return false;
        }));
```

Ejemplo de Php:

```
$client = new Memcached;
$client->setOption(Memcached::OPT_CLIENT_MODE, Memcached::DYNAMIC_CLIENT_MODE);
$client->addServer("xxxx", 11211);

$nodes = [];
$target_ips_count = 0;
do {
    # The PHP memcached client only updates the server list if the polling interval has
 expired and a
    # command is sent
    $client->get('test');

    $nodes = $client->getServerList();

    sleep(1);
    $target_ips_count = 0;

    // For IPv4 use FILTER_FLAG_IPV4
    $target_ips_count = count(array_filter($nodes, function($node) { return
 filter_var($node["ipaddress"], FILTER_VALIDATE_IP, FILTER_FLAG_IPV6); }));

} while (count($nodes) !== $target_ips_count);
```

Todas las conexiones de cliente existentes que se crearon antes de que se actualizara la detección de IP seguirán conectadas mediante el protocolo anterior. Todos los clientes validados se volverán a conectar automáticamente al clúster mediante el nuevo protocolo IP una vez que se detecten los cambios en el resultado de los comandos de detección del clúster. Sin embargo, esto depende de la implementación del cliente.

Clústeres de ElastiCache de doble pila compatibles con TLS

Cuando se habilita TLS para los clústeres de ElastiCache, las funciones de detección de clústeres (`cluster slots`, `cluster shards` y `cluster nodes` para Redis) o `config get cluster` para Memcached devuelven nombres de host en lugar de IP. A continuación, se utilizan los nombres de host en lugar de las IP para conectarse al clúster de ElastiCache y realizar un protocolo de enlace TLS. Esto significa que los clientes no se verán afectados por el parámetro de detección de IP. En el caso de los clústeres habilitados para TLS, el parámetro de detección de IP no tiene ningún efecto en el protocolo IP preferido. En cambio, el protocolo IP utilizado se determinará según el protocolo IP que prefiera el cliente al resolver los nombres de host de DNS.

Clientes de Java

Al conectarse desde un entorno de Java que admite IPv4 e IPv6, Java preferirá de forma predeterminada IPv4 en lugar de IPv6 por motivos de compatibilidad con versiones anteriores. Sin embargo, la preferencia del protocolo IP se puede configurar mediante los argumentos de JVM. Para preferir IPv4, JVM acepta `-Djava.net.preferIPv4Stack=true` y para preferir IPv6 establece `-Djava.net.preferIPv6Stack=true`. La configuración de `-Djava.net.preferIPv4Stack=true` significa que JVM ya no realizará ninguna conexión IPv6. En el caso de Valkey o Redis OSS, esto incluye a otras aplicaciones que no son de Valkey ni de Redis OSS.

Preferencias de nivel de host

En general, si el cliente o el entorno de ejecución del cliente no ofrecen opciones de configuración para establecer una preferencia de protocolo IP, al realizar la resolución de DNS, el protocolo IP dependerá de la configuración del host. De forma predeterminada, la mayoría de los hosts prefieren IPv6 en lugar de IPv4, pero esta preferencia se puede establecer en el nivel de host. Esto afectará a todas las solicitudes de DNS de ese host, no solo a las dirigidas a los clústeres de ElastiCache.

Hosts de Linux

Para Linux, se puede configurar una preferencia de protocolo IP modificando el archivo `gai.conf`. El archivo `gai.conf` se encuentra en `/etc/gai.conf`. Si no se especifica `gai.conf`, debería haber disponible un ejemplo en `/usr/share/doc/glibc-common-x.xx/gai.conf` que se pueda copiar a `/etc/gai.conf`. Además, la configuración predeterminada no debe estar comentada. Si desea actualizar la configuración para preferir IPv4 al conectarse a un clúster de ElastiCache, actualice la prioridad del rango de CIDR que abarca las IP del clúster para que esté por encima de la prioridad de las conexiones IPv6 predeterminadas. De forma predeterminada, las

conexiones IPv6 tienen una prioridad de 40. Por ejemplo, suponiendo que el clúster esté ubicado en una subred con el CIDR 172.31.0.0:0/16, la siguiente configuración haría que los clientes prefirieran las conexiones IPv4 a ese clúster.

```
label ::1/128       0
label ::/0          1
label 2002::/16     2
label ::/96         3
label ::ffff:0:0/96 4
label fec0::/10     5
label fc00::/7      6
label 2001:0::/32   7
label ::ffff:172.31.0.0/112 8
#
#    This default differs from the tables given in RFC 3484 by handling
#    (now obsolete) site-local IPv6 addresses and Unique Local Addresses.
#    The reason for this difference is that these addresses are never
#    NATed while IPv4 site-local addresses most probably are.  Given
#    the precedence of IPv6 over IPv4 (see below) on machines having only
#    site-local IPv4 and IPv6 addresses a lookup for a global address would
#    see the IPv6 be preferred.  The result is a long delay because the
#    site-local IPv6 addresses cannot be used while the IPv4 address is
#    (at least for the foreseeable future) NATed.  We also treat Teredo
#    tunnels special.
#
# precedence  <mask>   <value>
#    Add another rule to the RFC 3484 precedence table.  See section 2.1
#    and 10.3 in RFC 3484.  The default is:
#
precedence  ::1/128       50
precedence  ::/0          40
precedence  2002::/16     30
precedence ::/96          20
precedence ::ffff:0:0/96  10
precedence ::ffff:172.31.0.0/112 100
```

Puede encontrar más información disponible sobre `gai.conf` en la página principal de Linux

Hosts de Windows

El proceso para los hosts de Windows es similar. Para los hosts de Windows puede ejecutar `netsh interface ipv6 set prefix CIDR_CONTAINING_CLUSTER_IPS PRECEDENCE LABEL`. Esto tiene el mismo efecto que modificar el archivo `gai.conf` en los hosts de Linux.

Esto actualizará las políticas de preferencias, de modo que se prefieran las conexiones IPv4 en lugar de las conexiones IPv6 para el rango de CIDR especificado. Por ejemplo, suponiendo que el clúster esté en una subred con el CIDR 172.31.0.0:0/16, ejecutar `netsh interface ipv6 set prefix ::ffff:172.31.0.0:0/112 100 15` generaría la siguiente tabla de prioridades, lo que haría que los clientes prefirieran IPv4 al conectarse al clúster.

```
C:\Users\Administrator>netsh interface ipv6 show prefixpolicies
Querying active state...

Precedence Label Prefix
---------- ----- --------------------------------
100 15 ::ffff:172.31.0.0:0/112
20 4 ::ffff:0:0/96
50 0 ::1/128
40 1 ::/0
30 2 2002::/16
5 5 2001::/32
3 13 fc00::/7
1 11 fec0::/10
1 12 3ffe::/16
1 3 ::/96
```

Administración de la memoria reservada para Valkey y Redis OSS

La memoria reservada es una memoria que se aparta del uso para los datos. Cuando se realiza una copia de seguridad o una conmutación por error, Valkey y Redis OSS usan la memoria disponible para registrar las operaciones de escritura en el clúster mientras los datos del clúster se escriben en el archivo .rdb. Si no dispone de suficiente memoria disponible para todas las operaciones de escritura, se produce un error en el proceso. A continuación se ofrece información sobre distintas opciones para administrar la memoria reservada para ElastiCache para Redis OSS y cómo aplicar dichas opciones.

Temas

- ¿Cuánta memoria reservada necesita?
- Parámetros de administración de la memoria reservada
- Especificación del parámetro de administración de memoria reservada

¿Cuánta memoria reservada necesita?

Si ejecuta una versión de Redis OSS anterior a la 2.8.22, reserve más memoria para las copias de seguridad y las conmutaciones por error que si ejecuta la versión 2.8.22 de Redis OSS o posteriores. Este requisito se debe a las distintas maneras que ElastiCache para Redis OSS implementa el proceso de copia de seguridad. La norma general es reservar la mitad del valor de `maxmemory` del tipo de nodo como capacidad adicional para Redis OSS en versiones anteriores a la 2.8.22 y un cuarto de este mismo valor para Redis OSS versión 2.8.22 y posteriores.

Debido a las diferentes formas en que ElastiCache implementa el proceso de copia de seguridad y replicación, la norma general es reservar el 25 % del valor de `maxmemory` de un tipo de nodo mediante el parámetro `reserved-memory-percent`. Este es el valor predeterminado y el valor recomendado para la mayoría de los casos.

Cuando los tipos de instancia micro y pequeñas ampliables funcionan cerca de los límites de `maxmemory`, es posible que se utilicen intercambios. Para mejorar la fiabilidad operativa en estos tipos de instancia durante las copias de seguridad, la replicación y los momentos de tráfico elevado, recomendamos aumentar el valor del parámetro `reserved-memory-percent` hasta un 30 % en los tipos de instancia pequeños y hasta un 50 % en los tipos de instancia micro.

Para cargas de trabajo con mucho volumen de escritura en clústeres de ElastiCache con organización de datos en niveles, recomendamos aumentar `reserved-memory-percent` hasta un 50 % de la memoria disponible del nodo.

Para obtener más información, consulte los siguientes temas:

- Forma de garantizar que dispone de memoria suficiente para crear una instantánea de Valkey o Redis OSS
- Cómo se implementan la sincronización y la copia de seguridad
- Organización de datos en niveles en ElastiCache

Parámetros de administración de la memoria reservada

Desde el 16 de marzo de 2017, Amazon ElastiCache proporciona dos parámetros mutuamente excluyentes para administrar la memoria de Valkey o Redis OSS: `reserved-memory` y `reserved-memory-percent`. Ninguno de estos parámetros forma parte de la distribución de Valkey o Redis OSS.

En función de cuándo haya comenzado a ser cliente de ElastiCache, uno de estos parámetros será el parámetro predeterminado de administración de la memoria. Este parámetro se aplica cuando crea un nuevo clúster o grupo de replicación de Valkey o Redis OSS y utiliza un grupo de parámetros predeterminado.

- Para clientes que comenzaron antes del 16 de marzo de 2017: cuando cree un clúster o grupo de replicación de Redis OSS mediante el grupo de parámetros predeterminado, el parámetro de administración de memoria es `reserved-memory`. En este caso, cero (0) bytes de memoria se encuentran reservados.
- Para clientes que comenzaron el 16 de marzo de 2017 o después de esa fecha: cuando cree un clúster o grupo de replicación de Valkey o Redis OSS mediante el grupo de parámetros predeterminado, el parámetro de administración de la memoria es `reserved-memory-percent`. En este caso, el 25 % del valor `maxmemory` del nodo se encuentra reservado para fines no relacionados con datos.

Después de leer sobre los dos parámetros de administración de la memoria de Valkey o Redis OSS, puede que prefiera utilizar el que no es el predeterminado o uno con valores no predeterminados. En este caso, puede cambiar al otro parámetro de administración de la memoria reservada.

Para cambiar el valor de ese parámetro, puede crear un grupo de parámetros personalizado y modificarlo a fin de utilizar el parámetro y el valor de administración de la memoria preferidos. A continuación, podrá usar el grupo de parámetros personalizado cada vez que cree un nuevo clúster o

grupo de replicación de Valkey o Redis OSS. Para los clústeres o grupos de reproducción existentes, puede modificarlos para que usen su grupo de parámetros personalizado.

Para obtener más información, consulte los siguientes temas:

- Especificación del parámetro de administración de memoria reservada
- Creación de un grupo de parámetros de ElastiCache
- Modificación de un grupo de parámetros de ElastiCache
- Modificación de un clúster de ElastiCache
- Modificación de un grupo de reproducción

Parámetro reserved-memory

Antes del 16 de marzo de 2017, toda la administración de la memoria reservada de ElastiCache para Redis OSS se realizaba con el parámetro `reserved-memory`. El valor predeterminado de `reserved-memory` es 0. Este valor predeterminado no reserva memoria alguna como capacidad adicional para Valkey o Redis OSS y permite que Valkey o Redis OSS consuma toda la memoria de un nodo con datos.

La modificación del parámetro `reserved-memory` para tener suficiente memoria disponible para las copias de seguridad y las conmutaciones por error requiere crear un grupo de parámetros personalizado. En este grupo de parámetros personalizado, el parámetro `reserved-memory` se establece en un valor adecuado a la versión de Valkey o Redis OSS que ejecuta en su clúster y al tipo de nodo del clúster. Para obtener más información, consulte ¿Cuánta memoria reservada necesita?

El parámetro `reserved-memory` es específico para ElastiCache y no forma parte de la distribución general de Redis OSS.

El procedimiento siguiente muestra cómo usar `reserved-memory` para administrar la memoria en su clúster de Valkey o Redis OSS.

Para reservar memoria mediante reserved-memory

1. Cree un grupo de parámetros personalizado especificando la familia del grupo de parámetros que coincide con la versión del motor que está ejecutando, como, por ejemplo, la familia del grupo de parámetros `redis2.8`. Para obtener más información, consulte Creación de un grupo de parámetros de ElastiCache.

```
aws elasticache create-cache-parameter-group \
   --cache-parameter-group-name redis6x-m3xl \
   --description "Redis OSS 2.8.x for m3.xlarge node type" \
   --cache-parameter-group-family redis6.x
```

2. Calcule el número de bytes de memoria que se deben reservar para la sobrecarga de Valkey o Redis OSS. Encontrará el valor de `maxmemory` para su tipo de nodo en Parámetros específicos de tipos de nodos de Redis OSS.

3. Modifique el grupo de parámetros personalizado de modo que el parámetro `reserved-memory` coincida con el número de bytes que calculó en el paso anterior. En el siguiente ejemplo de la AWS CLI, se da por sentado que ejecuta una versión de Redis OSS anterior a la 2.8.22 y que necesita reservar la mitad del valor de `maxmemory` del nodo. Para obtener más información, consulte Modificación de un grupo de parámetros de ElastiCache.

   ```
   aws elasticache modify-cache-parameter-group \
      --cache-parameter-group-name redis28-m3xl \
      --parameter-name-values "ParameterName=reserved-memory,
    ParameterValue=7130316800"
   ```

   Necesita un grupo de parámetros personalizado individual para cada tipo de nodo que utilice, ya que cada tipo de nodo tiene un valor de `maxmemory` distinto. Por lo tanto, cada tipo de nodo necesita un valor diferente para `reserved-memory`.

4. Modifique el clúster o grupo de replicación de Redis OSS de forma que use su grupo de parámetros personalizado.

   En el ejemplo de la CLI siguiente se modifica el clúster `my-redis-cluster` de forma que comience a usar de inmediato el grupo de parámetros personalizado `redis28-m3xl`. Para obtener más información, consulte Modificación de un clúster de ElastiCache.

   ```
   aws elasticache modify-cache-cluster \
      --cache-cluster-id my-redis-cluster \
      --cache-parameter-group-name redis28-m3xl \
      --apply-immediately
   ```

   En el ejemplo de la CLI siguiente se modifica el grupo de reproducción `my-redis-repl-grp` de forma que comience a usar de inmediato el grupo de parámetros personalizado `redis28-m3xl`. Para obtener más información, Modificación de un grupo de reproducción.

```
aws elasticache modify-replication-group \
   --replication-group-id my-redis-repl-grp \
   --cache-parameter-group-name redis28-m3xl \
   --apply-immediately
```

Parámetro reserved-memory-percent

El 16 de marzo de 2017, Amazon ElastiCache presentó el parámetro `reserved-memory-percent` y ahora se encuentra disponible en todas las versiones de ElastiCache para Redis OSS. El propósito del parámetro `reserved-memory-percent` es simplificar la administración de la memoria reservada en todos los clústeres. Esta simplificación se consigue al disponer de un único grupo de parámetros para cada familia de grupos de parámetros (como `redis2.8`) para administrar la memoria reservada de sus clústeres, sea cual sea el tipo de nodo. El valor de predeterminado para `reserved-memory-percent` es 25 (25 por ciento).

El parámetro `reserved-memory-percent` es específico para ElastiCache y no forma parte de la distribución general de Redis OSS.

Si el clúster utiliza un tipo de nodo de la familia r6gd y el uso de memoria alcanza el 75 por ciento, la organización de datos en niveles se activará automáticamente. Para obtener más información, consulte Organización de datos en niveles en ElastiCache.

Para reservar memoria mediante reserved-memory-percent

Para utilizar `reserved-memory-percent` a fin de administrar la memoria en su clúster de ElastiCache para Redis OSS, realice una de las acciones siguientes:

- Si ejecuta Redis OSS 2.8.22 o versiones posteriores, asigne el grupo de parámetros predeterminado a su clúster. El valor predeterminado de 25 debe ser suficiente. De lo contrario, siga los pasos que se describen a continuación para cambiar el valor.
- Si ejecuta una versión de Redis OSS anterior a la 2.8.22, probablemente tendrá que reservar más memoria que el 25 % predeterminado de `reserved-memory-percent`. Para ello, siga el procedimiento que se indica a continuación.

Para cambiar el valor porcentual de reserved-memory-percent

1. Cree un grupo de parámetros personalizado especificando la familia del grupo de parámetros que coincide con la versión del motor que está ejecutando, como, por ejemplo, la familia del

grupo de parámetros `redis2.8`. Necesita disponer de un grupo de parámetros personalizado porque no es posible modificar grupos de parámetros predeterminados. Para obtener más información, consulte Creación de un grupo de parámetros de ElastiCache.

```
aws elasticache create-cache-parameter-group \
   --cache-parameter-group-name redis28-50 \
   --description "Redis OSS 2.8.x 50% reserved" \
   --cache-parameter-group-family redis2.8
```

Dado que `reserved-memory-percent` se reserva la memoria como un porcentaje del valor `maxmemory` de un nodo, no necesita un grupo de parámetros personalizado para cada tipo de nodo.

2. Modifique el grupo de parámetros personalizados de modo que el parámetro `reserved-memory-percent` tenga el valor 50 (50 %). Para obtener más información, consulte Modificación de un grupo de parámetros de ElastiCache.

```
aws elasticache modify-cache-parameter-group \
   --cache-parameter-group-name redis28-50 \
   --parameter-name-values "ParameterName=reserved-memory-percent,
 ParameterValue=50"
```

3. Use este grupo de parámetros personalizado para todos los clústeres o grupos de replicación de Redis OSS que ejecuten una versión de Redis OSS anterior a la 2.8.22.

   En el ejemplo de la CLI siguiente se modifica el clúster de Redis OSS `my-redis-cluster` de forma que comience a usar de inmediato el grupo de parámetros personalizado `redis28-50`. Para obtener más información, consulte Modificación de un clúster de ElastiCache.

```
aws elasticache modify-cache-cluster \
   --cache-cluster-id my-redis-cluster \
   --cache-parameter-group-name redis28-50 \
   --apply-immediately
```

   En el ejemplo de la CLI siguiente se modifica el grupo de replicación de Redis OSS `my-redis-repl-grp` de forma que comience a usar de inmediato el grupo de parámetros personalizado `redis28-50`. Para obtener más información, consulte Modificación de un grupo de reproducción.

```
aws elasticache modify-replication-group \
```

```
    --replication-group-id my-redis-repl-grp \
    --cache-parameter-group-name redis28-50 \
    --apply-immediately
```

Especificación del parámetro de administración de memoria reservada

Si ya era un cliente de ElastiCache el 16 de marzo de 2017, su parámetro predeterminado de administración de memoria reservada es `reserved-memory` con cero (0) bytes de memoria reservada. Si comenzó a ser cliente de ElastiCache después del 16 de marzo de 2017, su parámetro predeterminado de administración de memoria reservada es `reserved-memory-percent` con un 25 % de la memoria reservada del nodo. Esto se cumple independientemente de cuándo haya creado el clúster o grupo de replicación de ElastiCache para Redis OSS. Sin embargo, puede cambiar el parámetro de administración de memoria reservada, mediante la AWS CLI o la API de ElastiCache.

Los parámetros `reserved-memory` y `reserved-memory-percent` son mutuamente excluyentes. Un grupo de parámetros siempre tiene un parámetro, pero nunca ambos. Puede cambiar el parámetro que usa el grupo de parámetros para la administración de memoria reservada modificando el grupo de parámetros. El grupo de parámetros debe ser personalizado, ya que no es posible modificar grupos de parámetros predeterminados. Para obtener más información, consulte Creación de un grupo de parámetros de ElastiCache.

Para especificar reserved-memory-percent

Para utilizar `reserved-memory-percent` como parámetro de administración de memoria reservada, modifique un grupo de parámetros personalizado mediante el comando `modify-cache-parameter-group`. Utilice el parámetro `parameter-name-values` para especificar `reserved-memory-percent` y un valor para ello.

El siguiente ejemplo de la CLI modifica el grupo de parámetros personalizados `redis32-cluster-on` para usar el parámetro `reserved-memory-percent` para administrar la memoria reservada. Se debe asignar un valor a `ParameterValue` para que el grupo de parámetros utilice el parámetro `ParameterName` a fin de administrar la memoria reservada. Para obtener más información, consulte Modificación de un grupo de parámetros de ElastiCache.

```
aws elasticache modify-cache-parameter-group \
   --cache-parameter-group-name redis32-cluster-on \
   --parameter-name-values "ParameterName=reserved-memory-percent, ParameterValue=25"
```

Para especificar reserved-memory

Para utilizar `reserved-memory` como parámetro de administración de memoria reservada, modifique un grupo de parámetros personalizado mediante el comando `modify-cache-parameter-group`. Utilice el parámetro `parameter-name-values` para especificar `reserved-memory` y un valor para ello.

El siguiente ejemplo de la CLI modifica el grupo de parámetros personalizados `redis32-m3xl` para usar el parámetro `reserved-memory` para administrar la memoria reservada. Se debe asignar un valor a `ParameterValue` para que el grupo de parámetros utilice el parámetro `ParameterName` a fin de administrar la memoria reservada. Dado que la versión del motor es posterior a la versión 2.8.22, establecemos el valor en `3565158400`, que es el 25 % del valor de `cache.m3.xlarge` de `maxmemory`. Para obtener más información, consulte Modificación de un grupo de parámetros de ElastiCache.

```
aws elasticache modify-cache-parameter-group \
   --cache-parameter-group-name redis32-m3xl \
   --parameter-name-values "ParameterName=reserved-memory, ParameterValue=3565158400"
```

Prácticas recomendadas al trabajar con clústeres basados en nodos de Valkey y Redis OSS

El uso de multi-AZ, disponer de suficiente memoria, cambiar el tamaño del clúster y minimizar el tiempo de inactividad son conceptos útiles que hay que tener en cuenta al trabajar con clústeres basados en nodos en Valkey o Redis OSS. Le recomendamos que revise y siga estos procedimientos recomendados.

Temas

- Minimización del tiempo de inactividad con Multi-AZ
- Forma de garantizar que dispone de memoria suficiente para crear una instantánea de Valkey o Redis OSS
- Cambio de tamaño de clústeres online
- Minimización del tiempo de inactividad durante el mantenimiento

Minimización del tiempo de inactividad con Multi-AZ

Existen varias instancias en las que ElastiCache Valkey o Redis OSS podrían tener que reemplazar un nodo principal. Entre estas se incluyen determinados tipos de mantenimiento planificado y el caso poco probable de que se produzca un error en el nodo principal o en la zona de disponibilidad.

Este reemplazo produce un tiempo de inactividad para el clúster, pero si Multi-AZ se encuentra habilitado, el tiempo de inactividad es mínimo. El rol del nodo primario tendrá una conmutación por error automática en una de las réplicas de lectura. No es necesario crear ni aprovisionar un nodo primario nuevo, ya que ElastiCache se encargará de esto de forma clara. Esta conmutación por error y promoción de réplica garantizan la posibilidad de reanudar la escritura en la réplica principal tan pronto como se complete la promoción.

Consulte Minimización del tiempo de inactividad en ElastiCache utilizando multi-AZ con Valkey y Redis OSS para obtener más información sobre multi-AZ y la minimización del tiempo de inactividad.

Forma de garantizar que dispone de memoria suficiente para crear una instantánea de Valkey o Redis OSS

Instantáneas y sincronizaciones en Valkey 7.2 y versiones posteriores y Redis OSS versión 2.8.22 y posteriores

Valkey admite de forma predeterminada instantáneas y sincronizaciones. Redis OSS 2.8.22 introduce un proceso de guardado sin ramificaciones que le permite asignar más memoria para el uso de su aplicación sin tener que aumentar el uso del espacio de intercambio durante las sincronizaciones y guardados. Para obtener más información, consulte Cómo se implementan la sincronización y la copia de seguridad.

Instantáneas y sincronizaciones de Redis OSS anteriores a la versión 2.8.22

Al trabajar con ElastiCache para Redis OSS, Redis OSS llama a un comando de escritura en segundo plano en diversos casos:

- Al crear una instantánea para una copia de seguridad.
- Al sincronizar réplicas con la instancia principal en un grupo de reproducción.
- Al habilitar la característica de archivo de solo anexado (AOF) para Redis OSS.
- Al promocionar una réplica al nodo primario (lo que produce la sincronización del nodo primario/réplica).

Siempre que Redis OSS ejecute un proceso de escritura en segundo plano, debe disponer de memoria suficiente para los procesos adicionales. Si no tiene suficiente memoria disponible, se produce un error en el proceso. Por ello, es importante elegir un tipo de instancia de nodo que tenga suficiente memoria al crear el clúster de Redis OSS.

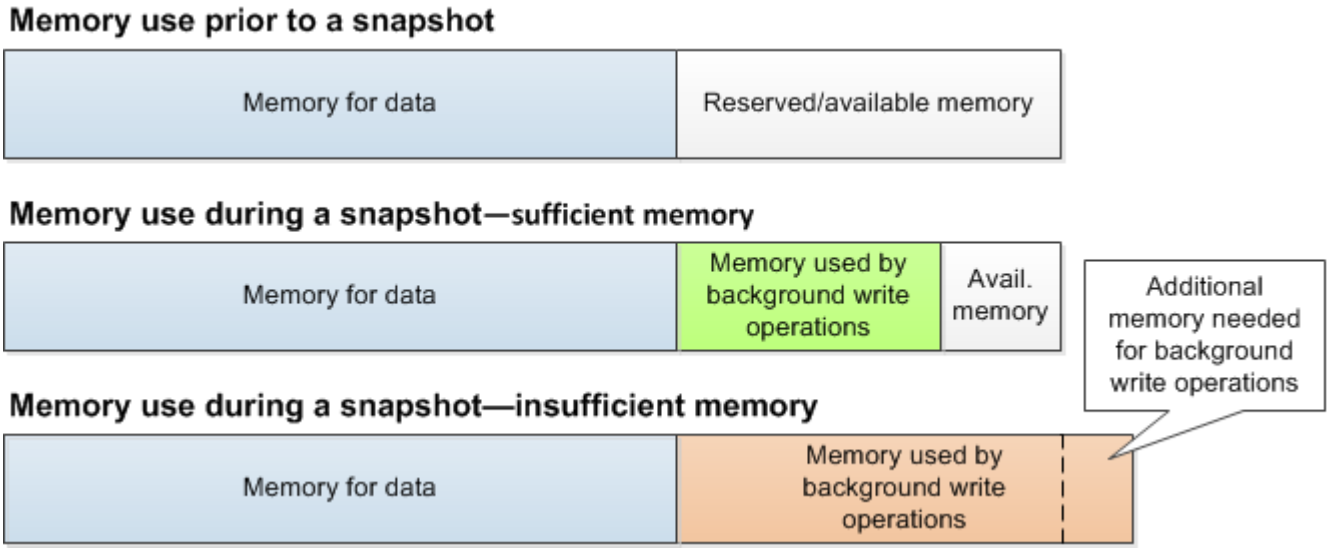
Proceso de escritura en segundo plano y uso de memoria con Valkey y Redis OSS

Siempre que se llame a un proceso de escritura en segundo plano, Valkey y Redis OSS ramifican su proceso (recuerde que estos motores son de un solo proceso). Una ramificación conserva sus datos en disco en un archivo de instantánea .rdb de Redis OSS. El resto de los servicios de la ramificación son todos operaciones de lectura y escritura. Para asegurarse de que la instantánea sea una instantánea puntual, todas las actualizaciones y adiciones se escriben en un área de la memoria disponible independiente del área de los datos.

Siempre que tenga suficiente memoria disponible para registrar todas las operaciones de escritura mientras se almacenan los datos en disco, no debería tener problemas de memoria insuficiente. Es probable que experimente problemas de memoria insuficiente si se da alguna de las siguientes condiciones:

- Su aplicación realiza muchas operaciones de escritura, lo que requiere una gran cantidad de memoria disponible para aceptar los datos nuevos o actualizados.
- Tiene muy poca memoria disponible en la que agregar datos nuevos o datos.
- Dispone de un gran conjunto de datos que requiere mucho tiempo para guardar en disco, lo que exige un gran número de operaciones de escritura.

El siguiente diagrama ilustra el uso de memoria al ejecutar un proceso de escritura en segundo plano.



Para obtener más información acerca del impacto en el rendimiento de hacer una copia de seguridad, consulte Impacto en el rendimiento de las copias de seguridad de los clústeres basados en nodos.

Para obtener más información acerca de cómo realizan Valkey y Redis OSS las instantáneas, consulte http://valkey.io.

Para obtener más información acerca de las regiones y zonas de disponibilidad, consulte Elección de regiones y zonas de disponibilidad para ElastiCache.

Evitar el agotamiento de la memoria al ejecutar una escritura en segundo plano

Siempre que se llame a un proceso de escritura en segundo plano, como `BGSAVE` o `BGREWRITEAOF`, para evitar que el proceso falle, debe tener más memoria disponible que la que consumirán las operaciones de escritura durante el proceso. Lo peor que podría ocurrir es que, durante la operación de escritura en segundo plano, se actualizasen todos los registros y se añadiesen algunos registros nuevos a la caché. Por ello, recomendamos que establezca `reserved-memory-percent` en 50 (50 %) en versiones de Redis OSS anteriores a 2.8.22, o bien en 25 (25 %) en Redis OSS 2.8.22 y versiones posteriores.

El valor `maxmemory` indica la memoria disponible para sus datos y para la carga adicional operativa. Puesto que no puede modificar el parámetro `reserved-memory` en el grupo de parámetros predeterminado, debe crear un grupo de parámetros personalizado para el clúster. El valor predeterminado para `reserved-memory` es 0, lo que permite a Redis OSS consumir la totalidad de maxmemory con datos, lo que posiblemente deja muy poca memoria para otros usos, como un

proceso de escritura en segundo plano. Para los valores `maxmemory` por tipo de instancia de nodo, consulte Parámetros específicos de tipos de nodos de Redis OSS.

También puede utilizar el parámetro `reserved-memory` para reducir la cantidad de memoria que se utiliza de forma predeterminada.

Para obtener más información sobre los parámetros específicos de Valkey y Redis en ElastiCache, consulte Parámetros de Valkey y Redis OSS.

Para obtener información acerca de cómo crear y modificar grupos de parámetros, consulte Creación de un grupo de parámetros de ElastiCache y Modificación de un grupo de parámetros de ElastiCache.

Cambio de tamaño de clústeres online

El cambio de las particiones implica agregar y eliminar particiones o nodos del clúster y redistribuir los espacios clave. Como resultado, varios aspectos influyen en la operación de cambio de las particiones, como la carga en el clúster, la utilización de memoria y el tamaño total de los datos. Para disfrutar de la mejor experiencia, recomendamos que siga las prácticas recomendadas de clúster global para una distribución uniforme del patrón de carga de trabajo. Además, recomendamos que siga los pasos que se detallan a continuación.

Antes de iniciar el cambio de las particiones, recomendamos lo siguiente:

- Probar la aplicación: si es posible, pruebe el comportamiento de la aplicación durante el cambio de las particiones en un entorno de ensayo.
- Recibir notificaciones anticipadas sobre problemas de escalado: el cambio de particiones es una operación que requiere mucho procesamiento. Por ello, recomendamos que mantenga el uso de la CPU por debajo del 80 por ciento en instancias de varios núcleos y en menos del 50 por ciento en instancias de un solo núcleo durante el cambio de particiones. Supervise las métricas de ElastiCache para Redis OSS e inicie el cambio de las particiones antes de que la aplicación comience a tener problemas de escalado. Las métricas de las que se puede realizar un seguimiento son `CPUUtilization`, `NetworkBytesIn`, `NetworkBytesOut`, `CurrConnections`, `NewConnections`, `FreeableMemory`, `SwapUsage` y `BytesUsedForCacheItems`.
- Comprobar que hay suficiente memoria libre disponible antes de la reducción horizontal: si va a realizar una reducción horizontal, asegúrese de que la memoria libre disponible en las particiones que se van a retener sea al menos 1,5 veces la memoria utilizada en las particiones que tiene previsto eliminar.

- Iniciar el cambio de las particiones durante las horas de menor actividad: esta práctica contribuye a reducir la latencia y el impacto en el rendimiento en el cliente durante la operación de cambio de las particiones. También ayuda a completar el cambio de las particiones con mayor rapidez ya que se pueden usar más recursos para la redistribución de ranuras.
- Revisar el comportamiento de tiempo de espera de cliente: es posible que algunos clientes observen una latencia más alta durante el cambio de tamaño del clúster en línea. La configuración de la biblioteca de cliente con un tiempo de espera más alto puede ayudar a conceder al sistema tiempo para conectar incluso en condiciones de carga más altas en servidor. En algunos casos, es posible que abra un gran número de conexiones al servidor. En estos casos, considere la posibilidad de agregar retardo exponencial a la lógica de reconexión. Si lo hace, puede ayudar a evitar que llegue una ráfaga de conexiones nuevas al servidor al mismo tiempo.
- Cargar las funciones en cada partición: al escalar horizontalmente el clúster, ElastiCache replicará automáticamente las funciones cargadas en uno de los nodos existentes (seleccionado de forma aleatoria) en los nuevos nodos. Si el clúster tiene Valkey 7.2 o una versión posterior o Redis OSS 7.0 o una versión posterior y la aplicación usa funciones, le recomendamos que cargue todas las funciones en todas las particiones antes de escalar horizontalmente para que el clúster no termine con diferentes funciones en distintas particiones.

Después del cambio de las particiones, tenga en cuenta lo siguiente:

- La reducción horizontal se puede realizar parcialmente si no hay suficiente memoria disponible en las particiones de destino. Si se produce este resultado, revise la memoria disponible y, si es necesario, reintente la operación. Los datos de las particiones de destino no se eliminarán.
- No se admiten los comandos `FLUSHALL` y `FLUSHDB` en los scripts Lua dentro durante una operación de cambio de particiones. En versiones anteriores a Redis OSS 6, no se admite el comando `BRPOPLPUSH` si opera en la ranura que se va a migrar.

Minimización del tiempo de inactividad durante el mantenimiento

La configuración de modo de clúster presenta la mejor disponibilidad durante operaciones administradas y no administradas. Se recomienda utilizar un cliente compatible con modo de clúster que se conecte al punto de enlace de detección de clústeres. Para el modo de clúster deshabilitado, se recomienda utilizar el punto de enlace principal para todas las operaciones de escritura.

Para la actividad de lectura, las aplicaciones pueden conectarse también a cualquier nodo del clúster. A diferencia del punto de conexión principal, los puntos de conexión de nodo se resuelven

en puntos de enlace específicos. Si realiza un cambio en su clúster, como añadir o eliminar una réplica, debe actualizar los puntos de enlace del nodo en su aplicación. Por eso, si el modo de clúster está deshabilitado, le recomendamos que utilice el punto de conexión del lector para la actividad de lectura.

Si la conmutación por error automática está habilitada en el clúster, es posible que el nodo principal cambie. Por lo tanto, la aplicación debe confirmar el rol del nodo y actualizar todos los puntos de enlace de lectura. Esta operación ayuda a garantizar que no esté provocando una carga importante en el nodo principal. Con la conmutación por error automática desactivada, el rol del nodo no cambia. Sin embargo, el tiempo de inactividad en las operaciones administradas o no administradas es mayor en comparación con los clústeres con conmutación por error automática habilitada.

Evite dirigir las solicitudes de lectura a un único nodo de réplica de lectura, ya que su falta de disponibilidad podría provocar una interrupción de la lectura. Opte por la lectura desde el servidor principal o asegúrese de tener al menos dos réplicas de lectura para evitar cualquier interrupción de la lectura durante el mantenimiento.

Estrategias de almacenamiento en caché para Memcached

En el siguiente tema, encontrará estrategias para completar y mantener la caché de Memcached.

Las estrategias que implemente para completar y mantener su caché dependen del tipo de datos que va a almacenar en su caché, así como de los patrones de acceso a dichos datos. Por ejemplo, probablemente no quiera utilizar la misma estrategia para una tabla de clasificación de los 10 mejores jugadores de un sitio de juegos y noticias de moda. En el resto de esta sección, se analizan las distintas estrategias comunes de mantenimiento de caché, junto con sus ventajas e inconvenientes.

Temas

- Réplicas de lectura
- Carga diferida
- Escritura indirecta
- Agregar TTL
- Temas relacionados

Réplicas de lectura

A menudo, el rendimiento de las cachés sin servidor de ElastiCache se puede mejorar considerablemente creando réplicas y leyendo desde ellas en lugar de desde el nodo de caché principal. Para obtener más información consulte () Prácticas recomendadas para utilizar réplicas de lectura.

Carga diferida

Como su nombre indica, la carga diferida es una estrategia de almacenamiento en caché que carga los datos en la caché solo cuando es necesario. Funciona como se describe a continuación.

Amazon ElastiCache es un almacén de valor de clave en memoria que se sitúa entre su aplicación y el almacén de datos (base de datos) al que accede. Siempre que su aplicación solicite datos, primero realizará una solicitud a la caché de ElastiCache. Si los datos existen en la caché y son actuales, ElastiCache devuelve los datos a su aplicación. Si los datos no existen en la caché o se han vencido, la aplicación solicita los datos del almacén de datos. A continuación, el almacén de datos devuelve los datos a su aplicación. Luego, su aplicación escribe los datos que recibió del almacén en la caché. De esta forma, se pueden recuperar más rápidamente la próxima vez que se los soliciten.

Un acierto de caché se produce cuando los datos se encuentran en la caché y no han vencido:

1. La aplicación solicita los datos a la caché.
2. La caché devuelve los datos a la aplicación.

Un error de caché se produce cuando los datos no se encuentran en la caché y han vencido:

1. La aplicación solicita los datos a la caché.
2. La caché no dispone de los datos solicitados, por lo que devuelve `null`.
3. La aplicación solicita los datos a la base de datos y los recibe.
4. La aplicación actualiza la caché con los datos nuevos.

Ventajas y desventajas de la carga diferida

Las ventajas de la carga diferida son las siguientes:

- Solo se almacenan en la caché los datos solicitados.

  Dado que nunca se solicita la mayoría de los datos, la carga diferida evita completar la caché con datos que no se solicitan.
- Los errores de nodo no son fatales para su aplicación.

  Cuando se produce un error en un nodo y se reemplaza por un nodo nuevo y vacío, la aplicación sigue funcionando, aunque con mayor latencia. Cuando se realizan solicitudes al nodo nuevo, cada error de caché da como resultado una consulta de la base de datos. Al mismo tiempo, la copia de datos se agrega a la caché para que las solicitudes posteriores se recuperen de la caché.

Las desventajas de la carga diferida son las siguientes:

- Existe una penalización de errores de caché. Cada error de caché genera tres acciones:
  1. Solicitud inicial de los datos a la caché
  2. Consulta de los datos en la base de datos
  3. Escritura de los datos en la caché

  Estos errores pueden provocar un retraso significativo en la obtención de los datos en la aplicación.

- Datos obsoletos.

  Si los datos se escriben en la caché solo cuando se produce un error de caché, los datos de la caché pueden quedar obsoletos. Este resultado se produce porque no hay actualizaciones en la caché cuando se cambian los datos en la base de datos. Para solucionar este problema, puede utilizar las estrategias Escritura indirecta y Agregar TTL.

Ejemplo de seudocódigo de carga diferida

El siguiente ejemplo es un seudocódigo de lógica de carga diferida.

```
// ****************************************
// function that returns a customer's record.
// Attempts to retrieve the record from the cache.
// If it is retrieved, the record is returned to the application.
// If the record is not retrieved from the cache, it is
//    retrieved from the database,
//    added to the cache, and
//    returned to the application
// ****************************************
get_customer(customer_id)

    customer_record = cache.get(customer_id)
    if (customer_record == null)

        customer_record = db.query("SELECT * FROM Customers WHERE id = {0}",
 customer_id)
        cache.set(customer_id, customer_record)

    return customer_record
```

Para este ejemplo, el código de aplicación que obtiene los datos es el siguiente.

```
customer_record = get_customer(12345)
```

Escritura indirecta

La estrategia de escritura indirecta agrega o actualiza los datos de la caché siempre que se escriben datos en la base de datos.

Ventajas y desventajas de la escritura indirecta

Las ventajas de la escritura indirecta son las siguientes:

- Los datos de la caché nunca quedan obsoletos.

  Dado que los datos de la caché se actualizan cada vez que se escriben en la base de datos, estos siempre se mantienen actualizados.
- Penalización de escritura frente a penalización de lectura.

  Toda operación de escritura implica dos acciones:

  1. Una operación de escritura en la caché
  2. Una operación de escritura en la base de datos

  Estas acciones añaden latencia al proceso. Dicho esto, los usuarios finales suelen ser más tolerantes con la latencia a la hora de actualizar datos que con la latencia a la hora de recuperar datos. Existe un sentido inherente que apunta a que las actualizaciones conllevan más trabajo y, por lo tanto, requieren mayor tiempo.

Las desventajas de la escritura indirecta son las siguientes:

- Pérdida de datos.

  Si pone en marcha un nodo nuevo, ya sea debido a un error de nodo o a una operación de escalado horizontal, existen datos que se perderán. Estos datos siguen faltando hasta que se agregan o actualizan en la base de datos. Puede minimizar esto al implementar una carga diferida con escritura indirecta.
- Pérdida de caché.

  La mayoría de los datos nunca se leen, lo cual es un desperdicio de recursos. Al agregar un valor de periodo de vida (TTL), puede minimizar el desperdicio de espacio.

Ejemplo de seudocódigo de escritura indirecta

El siguiente ejemplo es un seudocódigo de lógica de escritura indirecta.

```
// ****************************************
// function that saves a customer's record.
// ****************************************
```

```
save_customer(customer_id, values)

    customer_record = db.query("UPDATE Customers WHERE id = {0}", customer_id, values)
    cache.set(customer_id, customer_record)
    return success
```

Para este ejemplo, el código de aplicación que obtiene los datos es el siguiente.

```
save_customer(12345,{"address":"123 Main"})
```

Agregar TTL

La carga diferida da lugar a que los datos queden obsoletos, pero no falla con nodos vacíos. La escritura diferida mantiene los datos siempre actualizados, pero puede fallar con nodos vacíos y puede llenar la caché con datos superfluos. Al agregar un valor de periodo de vida (TTL) a cada escritura, puede tener las ventajas de cada estrategia. Al mismo tiempo, puede evitar en gran medida saturar la memoria caché con datos adicionales.

El periodo de vida (TTL) es un valor entero que especifica el número de segundos hasta que caduque la clave. Valkey o Redis OSS puede especificar segundos o milisegundos para este valor. Memcached especifica este valor en segundos. Cuando una aplicación intenta leer una clave vencida, la trata como si no se encontrara la clave. La base de datos se consulta para la clave y se actualiza la caché. Este enfoque no garantiza que un valor no se encuentre obsoleto. Sin embargo, evita que los datos queden demasiado obsoletos y se actualizan los valores de la caché con frecuencia desde la base de datos.

Para obtener más información, consulte Valkey and Redis OSS commands o los comandos de Memcached`set`.

Ejemplos de seudocódigo de TTL

El siguiente ejemplo es un seudocódigo de lógica de escritura indirecta con TTL.

```
// ****************************************
// function that saves a customer's record.
// The TTL value of 300 means that the record expires
//    300 seconds (5 minutes) after the set command
//    and future reads will have to query the database.
// ****************************************
save_customer(customer_id, values)
```

```
    customer_record = db.query("UPDATE Customers WHERE id = {0}", customer_id, values)
    cache.set(customer_id, customer_record, 300)

    return success
```

El siguiente ejemplo es un seudocódigo de lógica de carga diferida con TTL.

```
// ***************************************
// function that returns a customer's record.
// Attempts to retrieve the record from the cache.
// If it is retrieved, the record is returned to the application.
// If the record is not retrieved from the cache, it is
//    retrieved from the database,
//    added to the cache, and
//    returned to the application.
// The TTL value of 300 means that the record expires
//    300 seconds (5 minutes) after the set command
//    and subsequent reads will have to query the database.
// ***************************************
get_customer(customer_id)

    customer_record = cache.get(customer_id)

    if (customer_record != null)
        if (customer_record.TTL < 300)
            return customer_record        // return the record and exit function

    // do this only if the record did not exist in the cache OR
    //    the TTL was >= 300, i.e., the record in the cache had expired.
    customer_record = db.query("SELECT * FROM Customers WHERE id = {0}", customer_id)
    cache.set(customer_id, customer_record, 300)  // update the cache
    return customer_record                // return the newly retrieved record and exit
 function
```

Para este ejemplo, el código de aplicación que obtiene los datos es el siguiente.

```
save_customer(12345,{"address":"123 Main"})
```

```
customer_record = get_customer(12345)
```

Temas relacionados

- Almacenamiento de datos en memoria
- Elección de un motor y una versión
- Escalado de ElastiCache

Administración del clúster basado en nodos en ElastiCache

ElastiCache ofrece dos opciones de implementación: caché sin servidor y clústeres basados en nodos. Cada una tiene sus propias capacidades y requisitos.

Esta sección contiene temas que lo ayudarán a administrar sus clústeres basados en nodos.

> **Note**
>
> Estos temas no son aplicables en ElastiCache Serverless.

Temas

- Clústeres de escalado automático de Valkey y Redis OSS
- Modificación del modo de clúster
- Replicación en regiones de AWS mediante los almacenes de datos globales
- Alta disponibilidad a través de grupos de reproducción
- Administración del mantenimiento de clústeres de ElastiCache
- Configuración de los parámetros de motor mediante grupos de parámetros de ElastiCache

Clústeres de escalado automático de Valkey y Redis OSS

Requisitos previos

El escalado automático de ElastiCache se limita a lo siguiente:

- Clústeres de Valkey o Redis OSS (modo de clúster habilitado) que ejecutan Valkey 7.2 y versiones posteriores o Redis OSS 6.0 y versiones posteriores
- Clústeres de organización de datos en niveles (modo de clúster habilitado) que ejecutan Valkey 7.2 y versiones posteriores o Redis OSS 7.0.7 y versiones posteriores

- Tamaños de instancia: Large, XLarge, 2XLarge
- Familias de tipos de instancia: R7g, R6g, R6gd, R5, M7g, M6g, M5, C7gn
- El escalado automático en ElastiCache no es compatible con clústeres que se ejecutan en almacenes de datos globales, Outposts o zonas locales.

Administración automática de la capacidad con el escalado automático de ElastiCache con Valkey o Redis OSS

El escalado automático de ElastiCache con Valkey o Redis OSS es la capacidad de aumentar o disminuir las particiones o réplicas deseadas en el servicio de ElastiCache de forma automática. ElastiCache utiliza el servicio de Auto Scaling de aplicaciones para proporcionar esta funcionalidad. Para obtener más información, consulte Auto Scaling de aplicaciones. Para utilizar el escalado automático, defina y aplique una política de escalado que utilice valores de destino y métricas de CloudWatch que asigne. El escalado automático de ElastiCache utiliza la política para aumentar o disminuir el número de instancias en respuesta a las cargas de trabajo reales.

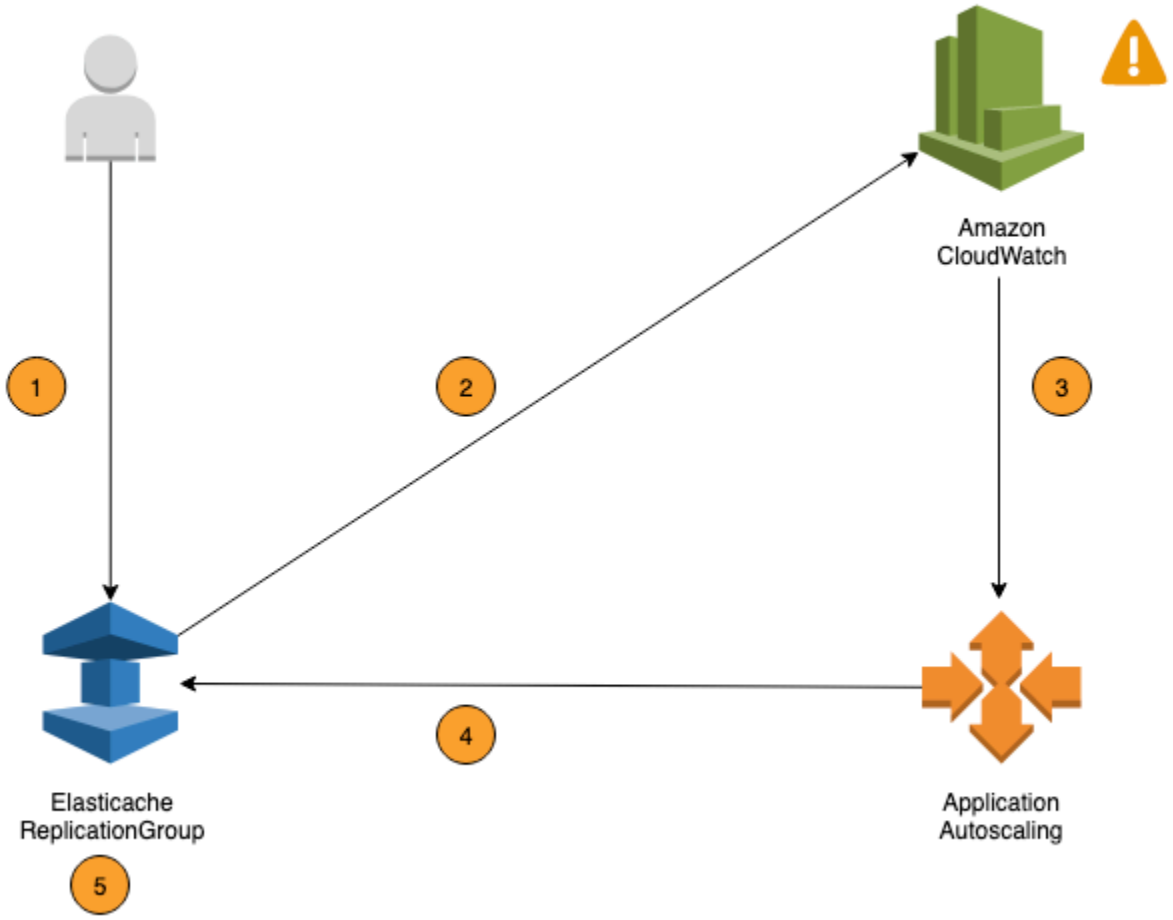
Puede usar la Consola de administración de AWS para aplicar una política de escalado en función de una métrica predefinida. Se define una `predefined metric` en una enumeración, de manera que puede especificarla por el nombre en el código o utilizarla en la Consola de administración de AWS. Las métricas personalizadas no se pueden seleccionar utilizando la Consola de administración de AWS. También puede utilizar la API de Auto Scaling de aplicaciones o la AWS CLI para aplicar una política de escalado basada en una métrica predefinida o personalizada.

ElastiCache para Valkey y Redis OSS admite el escalado para las siguientes dimensiones:

- Particiones: agregar/eliminar de forma automática particiones en el clúster de forma similar al cambio de particiones manual en línea. En este caso, el escalado automático de ElastiCache desencadena el escalado en su nombre.
- Réplicas: agregar/eliminar de forma automática réplicas en el clúster de manera similar a las operaciones manuales de aumento/reducción de réplicas. El escalado automático de ElastiCache para Valkey y Redis OSS añade o elimina las réplicas de forma uniforme en todas las particiones del clúster.

ElastiCache para Valkey y Redis OSS admite los siguientes tipos de políticas de escalado automático:

- Políticas de escalado de seguimiento de destino: permite aumentar o reducir el número de particiones/réplicas que ejecuta el servicio en función del valor de destino para una métrica determinada. Se asemeja a los termostatos que se utilizan para mantener la temperatura del hogar. Se selecciona una temperatura y el termostato hace el resto.
- Escalado programado para su aplicación. El escalado automático de ElastiCache para Valkey y Redis OSS permite aumentar o reducir el número de particiones o réplicas que ejecuta su servicio en función de la fecha y la hora.



En los siguientes pasos, se resume el proceso de escalado automático de ElastiCache para Valkey y Redis OSS, tal como se muestra en el diagrama anterior:

1. Crea una política de escalado automático de ElastiCache para su grupo de replicación.

2. El escalado automático de ElastiCache crea un par de alarmas de CloudWatch en su nombre. Cada par representa los límites superiores e inferiores de las métricas. Estas alarmas de CloudWatch se activan cuando la utilización real del clúster se desvía del objetivo de utilización durante un periodo de tiempo prolongado. Puede ver las alarmas en la consola.
3. Si el valor de la métrica configurado supera el objetivo de utilización (o no lo alcanza) durante un periodo específico, CloudWatch activa una alarma que invoca el escalado automático para evaluar la política de escalado.
4. El escalado automático de ElastiCache emite una solicitud de modificación para ajustar la capacidad del clúster.
5. ElastiCache procesa la solicitud de modificación, al aumentar (o reducir) dinámicamente la capacidad de las particiones o las réplicas del clúster para que sea similar a su uso objetivo.

Para comprender cómo funciona el escalado automático de ElastiCache, supongamos que tiene un clúster denominado `UsersCluster`. Mediante el monitoreo de las métricas de CloudWatch para `UsersCluster`, determinará el máximo de particiones que requiere el clúster cuando el tráfico se encuentra en su punto máximo, y el mínimo de particiones cuando el tráfico se encuentra en su punto más bajo. También decide un valor objetivo para la utilización de la CPU para el clúster `UsersCluster`. El escalado automático de ElastiCache utiliza el algoritmo de seguimiento de destino para asegurarse de que las particiones aprovisionadas de `UsersCluster` se ajusten de acuerdo a las necesidades, de tal forma que la utilización permanezca en el valor de objetivo o próxima a él.

> **Note**
>
> El escalado puede tardar un tiempo notable y requerirá recursos adicionales del clúster para que las particiones se reequilibren. El escalado automático de ElastiCache modifica la configuración de los recursos cuando la carga de trabajo real se mantiene elevada (o reducida) durante un periodo sostenido de varios minutos. El algoritmo de seguimiento de destino del escalado automático intenta mantener el objetivo de utilización en el valor elegido o en valores próximos a él a largo plazo.

Políticas de Auto Scaling

Una directiva de escalado tiene los siguientes componentes:

- Una métrica de destino: es la métrica de CloudWatch que utiliza el escalado automático de ElastiCache para Valkey y Redis OSS para determinar cuándo escalar y en qué medida.
- Capacidad mínima y máxima: el número mínimo y máximo de particiones o réplicas que se utilizarán para el escalado.

> **Important**
>
> Al crear la política de Auto Scaling, si la capacidad actual es superior a la capacidad máxima configurada, reducimos verticalmente hasta estar en la capacidad máxima durante la creación de la política. Del mismo modo, si la capacidad actual es inferior a la capacidad mínima configurada, escalamos horizontalmente hasta la capacidad mínima.

- Un periodo de recuperación: la cantidad de tiempo, en segundos, tras completarse una actividad de reducción o escalado horizontal antes de que pueda comenzar otra actividad de escalado horizontal.
- Un rol vinculado a servicios: un rol de AWS Identity and Access Management (IAM) que se encuentra vinculado a un servicio de AWS específico. Un rol vinculado a servicios incluye todos los permisos que requiere el servicio para llamar a otros servicios de AWS en su nombre. El escalado automático de ElastiCache genera este rol, `AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG`, de forma automática en su nombre.
- Habilitar o deshabilitar actividades de reducción horizontal: permite habilitar o deshabilitar actividades de reducción horizontal para una política.

Temas

- Métrica de destino para Auto Scaling
- Capacidad mínima y máxima
- Periodo de recuperación
- Activar o desactivar actividades de escalado descendente

Métrica de destino para Auto Scaling

En este tipo de política, una métrica predefinida o personalizada y un valor de destino de la métrica se especifica en una configuración de la política de escalado de seguimiento de destino. El escalado automático de ElastiCache para Valkey y Redis OSS crea y administra las alarmas de CloudWatch

que desencadenan la política de escalado y calcula el ajuste de escalado en función de la métrica y el valor objetivo. La política de escalado agrega o elimina las particiones/réplicas en función de las necesidades para mantener la métrica en el valor objetivo especificado o en un valor próximo. Además de mantener la métrica próxima al valor de destino, la política de escalado de seguimiento de destino también se ajusta a las fluctuaciones de la métrica producidas por una carga de trabajo en constante cambio. Esta política también minimiza las fluctuaciones rápidas del número de particiones/réplicas disponibles de su clúster.

Por ejemplo, considere una política de escalado que utilice la métrica `ElastiCachePrimaryEngineCPUUtilization` media predefinida. Esta política puede mantener la utilización de la CPU en el porcentaje de utilización especificado o en un valor próximo, como el 70 %.

> **Note**
>
> Para cada clúster, puede crear solo una política de Auto Scaling para cada métrica de destino.

Capacidad mínima y máxima

Particiones

Puede especificar el número máximo de particiones que se puede escalar mediante el escalado automático de ElastiCache para Valkey y Redis OSS. Este valor debe ser igual o inferior a 250 con un mínimo de 1. También puede especificar el número mínimo de particiones que va a administrar por medio del escalado automático. Este valor debe ser al menos 1, e igual o inferior al valor especificado para el máximo de 250 particiones.

Réplicas

Puede especificar el número máximo de réplicas que va a administrar por medio del escalado automático de ElastiCache para Valkey y Redis OSS. Este valor debe ser igual o inferior a 5. También puede especificar el número mínimo de réplicas que va a administrar mediante el escalado automático. Este valor debe ser al menos 1, e igual o inferior al valor especificado para el máximo de 5 réplicas.

A fin de determinar el número mínimo y máximo de particiones/réplicas que necesite para el tráfico normal, pruebe la configuración de Auto Scaling con la tasa de tráfico esperada para su modelo.

> **Note**
>
> Las políticas de escalado automático de ElastiCache aumentan la capacidad del clúster hasta que alcanza el tamaño máximo definido o hasta que se apliquen los límites de servicio. Para solicitar un aumento del límite, consulte AWS Service Limits y elija el tipo de límite Nodes per cluster per instance type (Nodos por clúster por tipo de instancias).

> **Important**
>
> La reducción horizontal sucede cuando no hay tráfico. Si el tráfico de una variante es cero, ElastiCache reduce horizontalmente de forma automática al número mínimo de instancias especificadas.

Periodo de recuperación

Puede ajustar la capacidad de respuesta de una política de escalado de seguimiento de destino al agregar periodos de recuperación que afecten al escalado de su clúster. Un periodo de recuperación bloquea solicitudes de escalado descendente o ascendente posteriores hasta que vence el periodo. Esto ralentiza las eliminaciones de particiones o réplicas del clúster de ElastiCache para Valkey y Redis OSS para solicitudes de reducción horizontal y la creación de particiones o réplicas para solicitudes de escalado horizontal. Puede especificar los siguientes periodos de recuperación:

- Una actividad de reducción horizontal reduce el número de particiones/réplicas del clúster. Un periodo de recuperación de escalado descendente especifica la cantidad de tiempo, en segundos, tras completarse una actividad de escalado descendente antes de que pueda comenzar otra actividad de escalado descendente.
- Una actividad de escalado horizontal incrementa el número de particiones/réplicas del clúster. Un periodo de recuperación de escalado ascendente especifica la cantidad de tiempo, en segundos, tras completarse una actividad de escalado ascendente antes de que pueda comenzar otra actividad de escalado ascendente.

Si no se especifica un periodo de recuperación de reducción o escalado horizontal, el valor predeterminado del escalado horizontal es de 600 segundos y de la reducción horizontal es de 900 segundos.

Activar o desactivar actividades de escalado descendente

Puede habilitar o deshabilitar actividades de escalado descendente para una política. La habilitación de actividades de reducción horizontal permite a la política de escalado eliminar particiones/réplicas. Al habilitarse actividades de escalado descendente, el periodo de recuperación de escalado descendente de la política de escalado se aplica a las actividades de escalado descendente. La deshabilitación de actividades de reducción horizontal impide a la política de escalado eliminar particiones/réplicas.

> **Note**
>
> Las actividades de escalado horizontal siempre se habilitan de modo que la política de escalado pueda crear particiones o réplicas de ElastiCache según sea necesario.

Permisos de IAM requeridos para el escalado automático

El escalado automático de ElastiCache para Valkey y Redis OSS es posible gracias a una combinación de API de ElastiCache, CloudWatch y Escalado automático de aplicaciones. Los clústeres se crean y actualizan con ElastiCache, las alarmas se crean con CloudWatch y las políticas de escalado se crean con Escalado automático de aplicaciones. Además de los permisos estándar de IAM para la creación y actualización de los clústeres, el usuario de IAM que accede a la configuración de escalado automático de ElastiCache debe tener los permisos adecuados para los servicios que admiten el escalado dinámico. En esta política más reciente, hemos añadido compatibilidad con el escalado vertical de Memcached con la acción `elasticache:ModifyCacheCluster`. Los usuarios de IAM deben contar con los permisos necesarios para utilizar las acciones que se muestran en la siguiente política de ejemplo:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:*",
                "elasticache:DescribeReplicationGroups",
                "elasticache:ModifyReplicationGroupShardConfiguration",
```

```
                "elasticache:IncreaseReplicaCount",
                "elasticache:DecreaseReplicaCount",
                "elasticache:DescribeCacheClusters",
                "elasticache:DescribeCacheParameters",
                "cloudwatch:DeleteAlarms",
                "cloudwatch:DescribeAlarmHistory",
                "cloudwatch:DescribeAlarms",
                "cloudwatch:DescribeAlarmsForMetric",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics",
                "cloudwatch:PutMetricAlarm",
                "cloudwatch:DisableAlarmActions",
                "cloudwatch:EnableAlarmActions",
                "iam:CreateServiceLinkedRole",
                "sns:CreateTopic",
                "sns:Subscribe",
                "sns:Get*",
                "sns:List*"
            ],
            "Resource": "arn:aws:iam::123456789012:role/autoscaling-roles-for-
cluster"
        }
    ]
}
```

Rol vinculado a servicios

El servicio de escalado automático de ElastiCache para Valkey y Redis OSS también necesita permiso para describir sus clústeres y las alarmas de CloudWatch, así como permisos para modificar la capacidad del objetivo de ElastiCache en su nombre. Si habilita el escalado automático para su clúster, crea un rol vinculado a servicios denominado `AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG`. Este rol vinculado a servicios concede al escalado automático de ElastiCache permiso para describir las alarmas de sus políticas y para supervisar y modificar la capacidad actual de la flota. El rol vinculado a servicios es el rol predeterminado para el escalado automático de ElastiCache. Para obtener más información, consulte Roles vinculados a servicios para Escalado automático de aplicaciones de ElastiCache para Redis OSS en la Guía del usuario de Escalado automático de aplicaciones.

Prácticas recomendadas de Auto Scaling

Antes de registrarse en Auto Scaling, recomendamos lo siguiente:

1. Utilizar solo una métrica de seguimiento: identifique si el clúster tiene CPU o cargas de trabajo intensivas de datos y utilice una métrica predefinida correspondiente para definir la política de escalado.
   - CPU del motor: `ElastiCachePrimaryEngineCPUUtilization` (dimensión de partición) o `ElastiCacheReplicaEngineCPUUtilization` (dimensión de réplica)
   - Uso de bases de datos: `ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage` Esta política de escalado funciona mejor si la política maxmemory está configurada como noeviction en el clúster.

   Le recomendamos que evite tener varias políticas por dimensión en el clúster. El escalado automático de ElastiCache para Valkey y Redis OSS realizará un escalado horizontal del destino escalable si alguna política de seguimiento de destino está lista para el escalado horizontal y solo realizará una reducción horizontal si todas las políticas de seguimiento de destino (que tengan habilitada la parte de reducción horizontal) están listas para la reducción horizontal. Si varias políticas indican al destino escalable que se realice una reducción o escalado horizontal al mismo tiempo, se escala en función de la política que proporciona la mayor capacidad para la reducción o el escalado horizontal.
2. Métricas personalizadas para el seguimiento de destinos: tenga cuidado al utilizar métricas personalizadas para el seguimiento de destinos, ya que el escalado automático es la característica más adecuada para el escalado horizontal en proporción a los cambios en las métricas elegidas para la política. Si esas métricas no cambian proporcionalmente a las acciones de escalado usadas para la creación de políticas, es posible que den lugar a acciones continuas de escalar horizontalmente o reducir horizontalmente que es posible que afecten a la disponibilidad o el costo.

   En el caso de los clústeres de datos por niveles (tipos de instancias de la familia r6gd), evite utilizar métricas basadas en memoria para escalar.
3. Escalado programado: si identifica que su carga de trabajo es determinista (alcance alto/bajo en un momento específico), recomendamos utilizar el escalado programado y configurar su capacidad objetivo según la necesidad. El seguimiento de destino es el más adecuado para las cargas de trabajo no determinista y para que el clúster funcione con la métrica de destino requerida al escalar horizontalmente cuando necesite más recursos y reducir horizontalmente cuando necesite menos.
4. Deshabilitar la reducción horizontal: el Auto Scaling del seguimiento de destino es el más adecuado para clústeres con aumento/disminución gradual de las cargas de trabajo, ya que

los picos o la disminución de las métricas pueden desencadenar oscilaciones consecutivas de reducción o escalado horizontal. Para evitar tales oscilaciones, puede comenzar con la reducción horizontal deshabilitada y luego siempre puede reducir horizontalmente de forma manual según sus necesidades.

5. Probar la aplicación: recomendamos que pruebe la aplicación con las cargas de trabajo mínimas y máximas estimadas a fin de determinar las particiones o réplicas mínimas y máximas necesarias para el clúster mientras crea políticas de escalado a fin de evitar problemas de disponibilidad. Auto Scaling puede escalar horizontalmente hasta el máximo y reducir horizontalmente en el umbral mínimo configurado para el destino.
6. Definición del valor de destino: puede analizar las métricas correspondientes de CloudWatch para la utilización del clúster durante un periodo de cuatro semanas a fin de determinar el umbral del valor objetivo. Si aún no se encuentra seguro de qué valor elegir, se recomienda comenzar con el valor mínimo admitido de métrica predefinida.
7. Auto Scaling del seguimiento de destino es la característica más adecuada para clústeres con distribución uniforme de las cargas de trabajo entre las dimensiones de particiones/réplicas. Tener una distribución no uniforme puede conducir a:
   - Escalado cuando no es necesario debido a un pico/caída de la carga de trabajo en algunas particiones o réplicas en almacenamiento activo.
   - Falta de escalado cuando sí es necesario debido a que la carga de trabajo media general se encuentra próxima al objetivo aunque tenga particiones o réplicas en el almacenamiento activo.

> **Note**
>
> Al escalar horizontalmente el clúster, ElastiCache replicará automáticamente las funciones cargadas en uno de los nodos existentes (seleccionadas de forma aleatoria) en los nuevos nodos. Si el clúster tiene Valkey o Redis 7.0 o una versión posterior y la aplicación usa funciones, le recomendamos que cargue todas las funciones en todas las particiones antes de escalar horizontalmente para que el clúster no termine con diferentes funciones en distintas particiones.

Después de registrarse en Auto Scaling, tenga en cuenta lo siguiente:

- Existen limitaciones en las configuraciones compatibles con Auto Scaling, por lo que recomendamos no cambiar la configuración de un grupo de reproducción que se haya registrado para Auto Scaling. A continuación se muestran algunos ejemplos:

- Modificación manual del tipo de instancias a tipos no admitidos.
- Asociación del grupo de reproducción a un almacén de datos global.
- Cambio del parámetro `ReservedMemoryPercent`.
- Aumento/disminución manual de particiones/réplicas más allá de la capacidad mínima y máxima configurada durante la creación de políticas.

Uso de Auto Scaling con particiones

Con el escalado automático de ElastiCache, puede utilizar políticas de seguimiento y programadas con su motor de Valkey o Redis OSS.

A continuación se proporcionan detalles sobre el seguimiento de destinos y las políticas programadas, así como sobre cómo aplicarlas mediante la Consola de administración de AWS, la AWS CLI y API.

Temas

- Políticas de escalado de seguimiento de destino
- Adición de una política de escalado
- Registro de un destino escalable
- Definición de una política de escalado
- Desactivación de actividad de escalado descendente
- Aplicación de una política de escalado
- Edición de una política de escalado
- Eliminación de una política de escalado
- Uso de CloudFormation para las políticas de Auto Scaling
- Escalado programado

Políticas de escalado de seguimiento de destino

Las políticas de escalado de seguimiento de destino le permiten seleccionar una métrica y establecer un valor de destino. El escalado automático de ElastiCache para Valkey y Redis OSS crea y administra las alarmas de CloudWatch que desencadenan la política de escalado y calcula el ajuste de escalado en función de la métrica y el valor de destino. La política de escalado agrega o elimina las particiones en función de las necesidades para mantener la métrica en el valor objetivo especificado o en un valor próximo. Además de mantener la métrica próxima al valor de destino, la

política de escalado de seguimiento de destino también se ajusta a las fluctuaciones de la métrica producidas por patrones de carga fluctuante y minimiza las fluctuaciones rápidas de la capacidad de la flota.

Por ejemplo, considere una política de escalado que utilice la métrica `ElastiCachePrimaryEngineCPUUtilization` media predefinida con el valor objetivo configurado. Esta política puede mantener la utilización de la CPU en el valor objetivo especificado o en un valor próximo.

Métricas predefinidas

Una métrica predefinida es una estructura que hace referencia a un nombre, una dimensión y una estadística (`average`) específicos de una métrica de CloudWatch determinada. La política de escalado automático define una de las siguientes métricas predefinidas para el clúster:

Nombre de métrica predefinido	Nombre de métrica de CloudWatch	Dimensión de métricas de CloudWatch	Tipos de instancia no aptos
`ElastiCachePrimaryEngineCPUUtilization`	`EngineCPUUtilization`	ReplicationGroupId, Role = Principal	Ninguno
`ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage`	`DatabaseCapacityUsageCountedForEvictPercentage`	Métricas de grupo de replicación de Valkey o Redis OSS	Ninguno
`ElastiCacheDatabaseMemoryUsageCountedForEvict`	`DatabaseMemoryUsageCountedForEvictPercentage`	Métricas de grupo de replicación de Valkey o Redis OSS	R6gd

Nombre de métrica predefinido	Nombre de métrica de CloudWatch	Dimensión de métricas de CloudWatch	Tipos de instancia no aptos
`Percentage`			

Los tipos de instancias con niveles de datos no pueden utilizar `ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage`, ya que estos tipos de instancias almacenan datos en la memoria y SSD. El caso de uso esperado para las instancias con niveles de datos es tener un uso de memoria del 100 por ciento y llenar el SSD según sea necesario.

Criterios de Auto Scaling para particiones

Cuando el servicio detecta que la métrica predefinida es igual o mayor que la configuración del objetivo, aumentará la capacidad de las particiones de forma automática. ElastiCache para Valkey y Redis OSS escala horizontalmente las particiones del clúster en una cantidad equivalente al mayor de dos números: la variación del porcentaje del objetivo y el 20 % de las particiones actuales. En el caso de la reducción horizontal, ElastiCache no se reduce horizontalmente de forma automática a menos que el valor total de la métrica sea inferior al 75 % del objetivo definido.

Para un ejemplo de escalado horizontal, si tiene 50 particiones y

- si su objetivo sufre una interrupción del 30 %, ElastiCache escala horizontalmente un 30 %, lo que da como resultado 65 particiones por clúster.
- si su objetivo sufre una interrupción del 10 %, ElastiCache escala horizontalmente de forma predeterminada un mínimo de 20 %, lo que da como resultado 60 particiones por clúster.

Para un ejemplo de reducción horizontal, si ha seleccionado un valor objetivo del 60 %, ElastiCache no se reducirá horizontalmente de forma automática hasta que la métrica sea inferior o igual al 45 % (25 % por debajo del objetivo de 60 %).

Consideraciones de Auto Scaling

Tenga en cuenta las siguientes consideraciones:

- En las políticas de escalado de seguimiento de destino, se presupone que el escalado ascendente se lleva a cabo cuando la métrica está por encima del valor de destino. No puede utilizar una

política de escalado de seguimiento de destinos si la métrica especificada está por debajo del valor de destino. ElastiCache para Valkey y Redis OSS escala horizontalmente las particiones con una desviación mínima del 20 % del objetivo de las particiones existentes en el clúster.

- Las políticas de escalado de seguimiento de destino no realizan el escalado cuando la métrica especificada no tiene datos suficientes. No realiza la reducción horizontalmente porque la carencia de datos no se interpreta como una infrautilización de recursos.
- Es posible que haya diferencias entre el valor de destino y los puntos de datos de la métrica real. Esto se debe a que el escalado automático de ElastiCache siempre actúa de forma conservadora y redondea hacia arriba o hacia abajo a la hora de determinar la cantidad de capacidad que se debe añadir o quitar. Con esto se evita que se agrega capacidad insuficiente o se elimine demasiada capacidad.
- Para garantizar la disponibilidad de la aplicación, el servicio se escala horizontalmente en proporción a la métrica tan rápido como puede, pero se reduce horizontalmente de forma más gradual.
- Puede tener varias políticas de escalado de seguimiento de destino en un clúster de ElastiCache para Valkey y Redis OSS, siempre que cada uno de ellos utilice una métrica diferente. El objetivo del escalado automático de ElastiCache siempre es dar prioridad a la disponibilidad, por lo que su comportamiento varía en función de si las políticas de seguimiento del objetivo están listas para el escalado o la reducción horizontales. Realizará un escalado horizontal del servicio si cualquiera de las políticas de seguimiento de destino está lista para el escalado horizontal, pero solo realizará la reducción horizontal si todas las políticas de seguimiento de destino (que tienen la parte de reducción horizontal habilitada) están listas para la reducción horizontal.
- No modifique ni elimine las alarmas de CloudWatch que administra el escalado automático de ElastiCache para las políticas de escalado de seguimiento de destino. El escalado automático de ElastiCache elimina de forma automática las alarmas cuando se elimina la política de escalado.
- El escalado automático de ElastiCache no le impide modificar de forma manual las particiones del clúster. Estos ajustes manuales no afectan a las alarmas de CloudWatch existentes adjuntas a la política de escalado, pero pueden afectar a las métricas que pueden activar estas alarmas de CloudWatch.
- Estas alarmas de CloudWatch administradas por Auto Scaling se definen a través de la métrica AVG (Promedio) en todas las particiones del clúster. Por lo tanto, tener particiones activas puede resultar en cualquiera de los siguientes escenarios:
  - escalado cuando no es necesario debido a la carga en algunas particiones activas que desencadenan una alarma de CloudWatch

  - no escalar cuando sí sea necesario debido a un AVG (Promedio) agregado en todas las particiones que causa que la alarma se interrumpa.
- Siguen aplicándose los límites predeterminados de ElastiCache en nodos por clúster. Por lo tanto, al optar por Auto Scaling y, si espera que los nodos máximos sean superiores al límite predeterminado, solicite un aumento del límite en AWS Service Limits y elija el tipo de límite Nodes per cluster per instance type (Nodos por clúster por tipo de instancias).
- Asegúrese de tener suficientes ENI (interfaces de red elásticas) disponibles en su VPC, que son necesarias durante el escalado horizontal. Para obtener más información, consulte Interfaces de red elástica.
- Si no hay suficiente capacidad disponible desde EC2, el escalado automático de ElastiCache no se escala en forma horizontal y se retrasa hasta que haya capacidad.
- El escalado automático de ElastiCache para Redis OSS durante la reducción horizontal no eliminará las particiones con ranuras que tengan un tamaño de elemento superior a 256 MB después de la serialización.
- Durante la reducción horizontal, no se eliminarán particiones si no se encuentra disponible la memoria suficiente en la configuración de particiones resultante.

Adición de una política de escalado

Puede agregar una política de escalado mediante la Consola de administración de AWS.

Cómo añadir una política de escalado automático a un clúster de ElastiCache para Valkey y Redis OSS

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Valkey o Redis OSS.
3. Elija el clúster al que desea agregar una política (elija el nombre del clúster y no el botón a su izquierda).
4. Elija la pestaña de Auto Scaling policies (Políticas de Auto Scaling).
5. Elija add dynamic scaling (agregar escalado dinámico).
6. En Policy name (Nombre de política), ingrese un nombre de política.
7. En Scalable Dimension (Dimensión escalable), elija shards (particiones).
8. Para la métrica de destino, elija una de las siguientes opciones:

- Primary CPU Utilization (Utilización de CPU principal) para crear una política basada en la utilización promedio de la CPU.
- Memory (Memoria) para crear una política basada en la memoria media de base de datos.
- Capacidad para crear una política basada en el uso promedio de la capacidad de la base de datos. La métrica de capacidad incluye la utilización de memoria y SSD para las instancias con niveles de datos y la utilización de la memoria para todos los demás tipos de instancias.

9. Para el valor objetivo, elija un valor mayor o igual a 35 y menor o igual a 70. El escalado automático mantendrá este valor para la métrica de destino seleccionada en las particiones de ElastiCache:

   - Utilización de la CPU principal: mantiene el valor objetivo de la métrica `EngineCPUUtilization` en los nodos principales.
   - Memoria: mantiene el valor objetivo de la métrica `DatabaseMemoryUsageCountedForEvictPercentage`
   - La capacidad mantiene el valor objetivo de la métrica `DatabaseCapacityUsageCountedForEvictPercentage`,

   Las particiones de clúster se agregan o quitan para mantener la métrica en un valor próximo al especificado.

10. (Opcional) La consola no admite periodos de recuperación de reducción o escalado horizontal. Utilice la AWS CLI para modificar los valores de periodo de recuperación.
11. En Capacidad mínima, escriba el número mínimo de particiones que debe mantener la política de escalado automático de ElastiCache.
12. En Capacidad máxima, escriba el número máximo de particiones que debe mantener la política de escalado automático de ElastiCache. Este valor debe ser igual o inferior a 250.
13. Seleccione Crear.

Registro de un destino escalable

Para poder utilizar el escalado automático con un clúster de ElastiCache para Valkey y Redis OSS, debe registrar su clúster en el escalado automático de ElastiCache. Esto se hace para definir la dimensión y los límites de escalado que se van a aplicar a ese clúster. El escalado automático de ElastiCache escala de forma dinámica el clúster a lo largo de la dimensión escalable

`elasticache:replication-group:NodeGroups`, que representa el número de particiones del clúster.

Uso de AWS CLI

Para registrar el clúster de ElastiCache para Valkey y Redis OSS, utilice el comando register-scalable-target con los siguientes parámetros:

- `--service-namespace` – establezca este valor en `elasticache`
- `--resource-id`: el identificador de recurso para el clúster. Para este parámetro, el tipo de recurso es `ReplicationGroup` y el identificador único es el nombre del clúster; por ejemplo, `replication-group/myscalablecluster`.
- `--scalable-dimension` – establezca este valor en `elasticache:replication-group:NodeGroups`.
- `--max-capacity` : el número máximo de particiones que administrará el escalado automático de ElastiCache. Para obtener información sobre la relación entre `--min-capacity`, `--max-capacity` y el número de particiones del clúster, consulte Capacidad mínima y máxima.
- `--min-capacity` : el número mínimo de particiones que administrará el escalado automático de ElastiCache. Para obtener información sobre la relación entre `--min-capacity`, `--max-capacity` y el número de particiones del clúster, consulte Capacidad mínima y máxima.

Example

En el siguiente ejemplo, registra un clúster de ElastiCache denominado `myscalablecluster`. El registro indica que el clúster debe escalarse de forma dinámica para tener de una a diez particiones.

Para Linux, macOS o Unix:

```
aws application-autoscaling register-scalable-target \
    --service-namespace elasticache \
    --resource-id replication-group/myscalablecluster \
    --scalable-dimension elasticache:replication-group:NodeGroups \
    --min-capacity 1 \
    --max-capacity 10 \
```

Para Windows:

```
aws application-autoscaling register-scalable-target ^
    --service-namespace elasticache ^
```

```
    --resource-id replication-group/myscalablecluster ^
    --scalable-dimension elasticache:replication-group:NodeGroups ^
    --min-capacity 1 ^
    --max-capacity 10 ^
```

Uso de la API

Para registrar el clúster de ElastiCache, utilice el comando register-scalable-target con los siguientes parámetros:

- ServiceNamespace: establezca este valor en elasticache.
- ResourceID: el identificador de recurso para el clúster de ElastiCache. Para este parámetro, el tipo de recurso es ReplicationGroup y el identificador único es el nombre del clúster; por ejemplo, `replication-group/myscalablecluster`.
- ScalableDimension: establezca este valor en `elasticache:replication-group:NodeGroups`.
- MinCapacity: el número mínimo de particiones que administrará el escalado automático de ElastiCache. Para obtener información sobre la relación entre --min-capacity, --max-capacity y el número de réplicas del clúster, consulte Capacidad mínima y máxima.
- MaxCapacity: el número máximo de particiones que administrará el escalado automático de ElastiCache. Para obtener información sobre la relación entre --min-capacity, --max-capacity y el número de réplicas del clúster, consulte Capacidad mínima y máxima.

Example

En el siguiente ejemplo, se registra un clúster de ElastiCache denominado `myscalablecluster` con la API de Escalado automático de aplicaciones. Este registro indica que el clúster debe escalarse de forma dinámica para tener de una a cinco réplicas.

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.RegisterScalableTarget
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
```

```
    "ServiceNamespace": "elasticache",
    "ResourceId": "replication-group/myscalablecluster",
    "ScalableDimension": "elasticache:replication-group:NodeGroups",
    "MinCapacity": 1,
    "MaxCapacity": 5
}
```

Definición de una política de escalado

Una configuración de la política de escalado de seguimiento de destino está representada por un bloque JSON en el que se definen las métricas y los valores de destino. Puede guardar una configuración de la política de escalado como bloque JSON en un archivo de texto. Puede utilizar ese archivo de texto al invocar AWS CLI o la API de Auto Scaling de aplicaciones. Para obtener más información sobre la sintaxis de configuración de políticas, consulte TargetTrackingScalingPolicyConfiguration en la Referencia de la API de Auto Scaling de aplicaciones.

Las siguientes opciones se encuentran disponibles para definir una configuración de la política de escalado de seguimiento de destino:

Temas

- Uso de una métrica predefinida
- Uso de una métrica personalizada
- Uso de periodos de recuperación

Uso de una métrica predefinida

Mediante las métricas predefinidas, puede definir rápidamente una política de escalado de seguimiento de destino para un clúster de ElastiCache para Valkey y Redis OSS que funciona con el seguimiento de destino en el escalado automático de ElastiCache.

Actualmente, ElastiCache admite las siguientes métricas predefinidas en el escalado automático de NodeGroup:

- ElastiCachePrimaryEngineCPUUtilization: el valor medio de la métrica `EngineCPUUtilization` en CloudWatch en todos los nodos principales del clúster.
- ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage: el valor medio de la métrica `DatabaseMemoryUsageCountedForEvictPercentage` en CloudWatch en todos los nodos principales del clúster.

- ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage: el valor medio de la métrica `ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage` en CloudWatch en todos los nodos principales del clúster.

Para obtener más información sobre las métricas de `EngineCPUUtilization`, `DatabaseMemoryUsageCountedForEvictPercentage` y `DatabaseCapaci` consulte Monitoreo del uso con métricas de CloudWatch. Para usar una métrica predefinida en su política de escalado, puede crear una configuración de seguimiento de destino para su política de escalado. Esta configuración debe incluir `PredefinedMetricSpecification` para la métrica predefinida y un TargetValue para el valor de destino de esa métrica.

Example

En el siguiente ejemplo, se describe una configuración de la política típica para el escalado de seguimiento de destino de un clúster de ElastiCache para Valkey y Redis OSS. En esta configuración, la métrica predefinida `ElastiCachePrimaryEngineCPUUtilization` se utiliza para ajustar el clúster en función de la utilización promedio de la CPU del 40 % en todos los nodos principales del clúster.

```
{
    "TargetValue": 40.0,
    "PredefinedMetricSpecification":
    {
        "PredefinedMetricType": "ElastiCachePrimaryEngineCPUUtilization"
    }
}
```

Uso de una métrica personalizada

Mediante las métricas personalizadas, puede definir una política de escalado de seguimiento de destino que cumpla sus requisitos personalizados. Puede definir una métrica personalizada en función de cualquier métrica de ElastiCache que cambie en proporción al escalado. No todas las métricas de ElastiCache funcionan para el seguimiento de destino. La métrica debe ser una métrica de utilización válida y describir el nivel de actividad de una instancia. El valor de la métrica debe aumentar o reducirse en proporción al número de particiones del clúster. Este aumento o reducción proporcionales son necesarios para utilizar los datos de las métricas a fin de reducir o escalar horizontalmente de forma proporcional el número de particiones.

Example

En el siguiente ejemplo se describe una configuración de seguimiento de destino para una política de escalado. En esta configuración, una métrica personalizada ajusta un clúster de ElastiCache para Redis OSS en función de una utilización promedio de la CPU del 50 % en todas las particiones de un clúster denominado `my-db-cluster`.

```
{
    "TargetValue": 50,
    "CustomizedMetricSpecification":
    {
        "MetricName": "EngineCPUUtilization",
        "Namespace": "AWS/ElastiCache",
        "Dimensions": [
            {
                "Name": "ReplicationGroup","Value": "my-db-cluster"
            },
            {
                "Name": "Role","Value": "PRIMARY"
            }
        ],
        "Statistic": "Average",
        "Unit": "Percent"
    }
}
```

Uso de periodos de recuperación

Puede especificar un valor, en segundos, a fin de que `ScaleOutCooldown` agregue un periodo de recuperación para el escalado horizontal de su clúster. De forma similar, puede agregar un valor, en segundos, a fin de que `ScaleInCooldown` agregue un periodo de recuperación para el escalado vertical de su clúster. Para obtener más información, consulte TargetTrackingScalingPolicyConfiguration en la Referencia de la API de Auto Scaling de aplicaciones.

En el siguiente ejemplo se describe una configuración de seguimiento de destino para una política de escalado. En esta configuración, la métrica predefinida `ElastiCachePrimaryEngineCPUUtilization` se utiliza para ajustar un clúster de ElastiCache para Redis OSS en función de la utilización promedio de la CPU del 40 % en todos los nodos principales de ese clúster. La configuración proporciona un periodo de recuperación de escalado descendente de 10 minutos y un periodo de recuperación de escalado ascendente de 5 minutos.

```
{
    "TargetValue": 40.0,
    "PredefinedMetricSpecification":
    {
        "PredefinedMetricType": "ElastiCachePrimaryEngineCPUUtilization"
    },
    "ScaleInCooldown": 600,
    "ScaleOutCooldown": 300
}
```

Desactivación de actividad de escalado descendente

Puede evitar que la configuración de la política de escalado de seguimiento de destino reduzca horizontalmente su clúster deshabilitando la actividad de reducción horizontal. La deshabilitación de la actividad de escalado horizontal evita que la política de escalado elimine particiones, a la vez que permite a la política de escalado crearlas según sea necesario.

Puede especificar un valor booleano a fin de que `DisableScaleIn` habilite o deshabilite la actividad de reducción horizontal para su clúster. Para obtener más información, consulte TargetTrackingScalingPolicyConfiguration en la Referencia de la API de Auto Scaling de aplicaciones.

En el siguiente ejemplo se describe una configuración de seguimiento de destino para una política de escalado. En esta configuración, la métrica predefinida `ElastiCachePrimaryEngineCPUUtilization` ajusta un clúster de ElastiCache para Valkey y Redis OSS en función de la utilización promedio de la CPU del 40 % en todos los nodos principales de ese clúster. La configuración deshabilita la actividad de escalado descendente para la política de escalado.

```
{
    "TargetValue": 40.0,
    "PredefinedMetricSpecification":
    {
        "PredefinedMetricType": "ElastiCachePrimaryEngineCPUUtilization"
    },
    "DisableScaleIn": true
}
```

Aplicación de una política de escalado

Tras registrar su clúster con escalado automático de ElastiCache para Valkey y Redis OSS y definir una política de escalado, puede aplicarla al clúster registrado. Para aplicar una política de escalado a un clúster de ElastiCache para Redis OSS, puede utilizar la AWS CLI o la API de Escalado automático de aplicaciones.

Aplicación de una política de escalado mediante la AWS CLI

Para aplicar una política de escalado a un clúster de ElastiCache para Valkey y Redis OSS, utilice el comando put-scaling-policy con los siguientes parámetros:

- --policy-name: el nombre de la política de escalado.
- --policy-type: establezca este valor en `TargetTrackingScaling`.
- --resource-id: el identificador del recurso. Para este parámetro, el tipo de recurso es `ReplicationGroup` y el identificador único es el nombre del clúster; por ejemplo, `replication-group/myscalablecluster`.
- --service-namespace: establezca este valor en `elasticache`.
- --scalable-dimension: establezca este valor en `elasticache:replication-group:NodeGroups`.
- --target-tracking-scaling-policy-configuration: la configuración de la política de escalado de seguimiento de destino que se utilizará para el clúster.

En el siguiente ejemplo, aplica una política de escalado de seguimiento de destino denominada `myscalablepolicy` a un clúster de ElastiCache para Valkey y Redis OSS denominado `myscalablecluster` con escalado automático de ElastiCache. Para ello, puede usar una configuración de la política guardada en un archivo denominado `config.json`.

Para Linux, macOS o Unix:

```
aws application-autoscaling put-scaling-policy \
    --policy-name myscalablepolicy \
    --policy-type TargetTrackingScaling \
    --resource-id replication-group/myscalablecluster \
    --service-namespace elasticache \
    --scalable-dimension elasticache:replication-group:NodeGroups \
    --target-tracking-scaling-policy-configuration file://config.json
```

Para Windows:

```
aws application-autoscaling put-scaling-policy ^
    --policy-name myscalablepolicy ^
    --policy-type TargetTrackingScaling ^
    --resource-id replication-group/myscalablecluster ^
    --service-namespace elasticache ^
    --scalable-dimension elasticache:replication-group:NodeGroups ^
    --target-tracking-scaling-policy-configuration file://config.json
```

Aplicación de una política de escalado mediante la API

Para aplicar una política de escalado a un clúster de ElastiCache para Valkey y Redis OSS, utilice el comando PutScalingPolicy de la AWS CLI con los siguientes parámetros:

- --policy-name: el nombre de la política de escalado.
- --resource-id: el identificador del recurso. Para este parámetro, el tipo de recurso es `ReplicationGroup` y el identificador único es el nombre del clúster; por ejemplo, `replication-group/myscalablecluster`.
- --service-namespace: establezca este valor en `elasticache`.
- --scalable-dimension: establezca este valor en `elasticache:replication-group:NodeGroups`.
- --target-tracking-scaling-policy-configuration: la configuración de la política de escalado de seguimiento de destino que se utilizará para el clúster.

En el siguiente ejemplo, aplica una política de escalado de seguimiento de destino denominada `myscalablepolicy` a un clúster de ElastiCache denominado `myscalablecluster` con el escalado automático de ElastiCache. Puede usar una configuración de la política en función de la métrica predefinida `ElastiCachePrimaryEngineCPUUtilization`.

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.PutScalingPolicy
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
```

```
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
    "PolicyName": "myscalablepolicy",
    "ServiceNamespace": "elasticache",
    "ResourceId": "replication-group/myscalablecluster",
    "ScalableDimension": "elasticache:replication-group:NodeGroups",
    "PolicyType": "TargetTrackingScaling",
    "TargetTrackingScalingPolicyConfiguration": {
        "TargetValue": 40.0,
        "PredefinedMetricSpecification":
        {
            "PredefinedMetricType": "ElastiCachePrimaryEngineCPUUtilization"
        }
    }
}
```

Edición de una política de escalado

Puede editar una política de escalado utilizando la Consola de administración de AWS, la AWS CLI o la API de Auto Scaling de aplicaciones.

Edición de una política de escalado mediante la Consola de administración de AWS

Cómo editar una política de escalado automático de un clúster de ElastiCache para Valkey y Redis OSS

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija el motor correspondiente.
3. Elija el clúster al que desea agregar una política (elija el nombre del clúster y no el botón a su izquierda).
4. Elija la pestaña de Auto Scaling policies (Políticas de Auto Scaling).
5. En Scaling policies (Políticas de escalado), elija el botón a la izquierda de la política de Auto Scaling que desea cambiar y, a continuación, elija Modify (Modificar).
6. Realice los cambios solicitados en la política.
7. Elija Modificar.

Edición de una política de escalado mediante la AWS CLI o la API

Puede utilizar AWS CLI o la API de Auto Scaling de aplicaciones para editar una política de escalado de la misma forma que aplica una política de escalado:

- Al usar la AWS CLI, especifique el nombre de la política que desea editar en el parámetro `--policy-name`. Especifique nuevos valores para los parámetros que desea cambiar.
- Al usar la API de Auto Scaling de aplicaciones, especifique el nombre de la política que desea editar en el parámetro `PolicyName`. Especifique nuevos valores para los parámetros que desea cambiar.

Para obtener más información, consulte Aplicación de una política de escalado.

Eliminación de una política de escalado

Puede eliminar una política de escalado utilizando la Consola de administración de AWS, AWS CLI o la API de Auto Scaling de aplicaciones.

Eliminación de una política de escalado mediante la Consola de administración de AWS

Cómo eliminar una política de escalado automático para un clúster de ElastiCache para Redis OSS

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Valkey o Redis OSS.
3. Elija el clúster con la política de Auto Scaling que desea editar (elija el nombre del clúster, no el botón a su izquierda).
4. Elija la pestaña de Auto Scaling policies (Políticas de Auto Scaling).
5. En Scaling policies (Políticas de escalado), elija la política de Auto Scaling y, a continuación, elija Delete (Eliminar).

Eliminación de una política de escalado mediante la AWS CLI

Para eliminar una política de escalado de un clúster de ElastiCache para Valkey y Redis OSS, utilice el comando delete-scaling-policy de la AWS CLI con los siguientes parámetros:

- --policy-name: el nombre de la política de escalado.

- --resource-id: el identificador del recurso. Para este parámetro, el tipo de recurso es `ReplicationGroup` y el identificador único es el nombre del clúster; por ejemplo, `replication-group/myscalablecluster`.
- --service-namespace: establezca este valor en `elasticache`.
- --scalable-dimension: establezca este valor en `elasticache:replication-group:NodeGroups`.

En el siguiente ejemplo, elimina una política de escalado de seguimiento de destino denominada `myscalablepolicy` de un clúster denominado `myscalablecluster`.

Para Linux, macOS o Unix:

```
aws application-autoscaling delete-scaling-policy \
    --policy-name myscalablepolicy \
    --resource-id replication-group/myscalablecluster \
    --service-namespace elasticache \
    --scalable-dimension elasticache:replication-group:NodeGroups
```

Para Windows:

```
aws application-autoscaling delete-scaling-policy ^
    --policy-name myscalablepolicy ^
    --resource-id replication-group/myscalablecluster ^
    --service-namespace elasticache ^
    --scalable-dimension elasticache:replication-group:NodeGroups
```

Eliminación de una política de escalado mediante la API

Para eliminar una política de escalado de un clúster de ElastiCache para Valkey y Redis OSS, utilice el comando DeleteScalingPolicy de la AWS CLI con los siguientes parámetros:

- --policy-name: el nombre de la política de escalado.
- --resource-id: el identificador del recurso. Para este parámetro, el tipo de recurso es `ReplicationGroup` y el identificador único es el nombre del clúster; por ejemplo, `replication-group/myscalablecluster`.
- --service-namespace: establezca este valor en `elasticache`.
- --scalable-dimension: establezca este valor en `elasticache:replication-group:NodeGroups`.

En el siguiente ejemplo, elimina una política de escalado de seguimiento de destino denominada `myscalablepolicy` de un clúster denominado `myscalablecluster`.

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.DeleteScalingPolicy
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
    "PolicyName": "myscalablepolicy",
    "ServiceNamespace": "elasticache",
    "ResourceId": "replication-group/myscalablecluster",
    "ScalableDimension": "elasticache:replication-group:NodeGroups"
}
```

Uso de CloudFormation para las políticas de Auto Scaling

Este fragmento muestra cómo crear una política de seguimiento de destino y aplicarla a un recurso AWS::ElastiCache::ReplicationGroup mediante el recurso AWS::ApplicationAutoScaling::ScalableTarget. Utiliza las funciones intrínsecas Fn::Join y Ref para construir la propiedad `ResourceId` con el nombre lógico del recurso `AWS::ElastiCache::ReplicationGroup` especificado en la misma plantilla.

```
ScalingTarget:
   Type: 'AWS::ApplicationAutoScaling::ScalableTarget'
   Properties:
     MaxCapacity: 3
     MinCapacity: 1
     ResourceId: !Sub replication-group/${logicalName}
     ScalableDimension: 'elasticache:replication-group:NodeGroups'
     ServiceNamespace: elasticache
     RoleARN: !Sub "arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/elasticache.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG"

  ScalingPolicy:
    Type: "AWS::ApplicationAutoScaling::ScalingPolicy"
    Properties:
```

```
      ScalingTargetId: !Ref ScalingTarget
      ServiceNamespace: elasticache
      PolicyName: testpolicy
      PolicyType: TargetTrackingScaling
      ScalableDimension: 'elasticache:replication-group:NodeGroups'
      TargetTrackingScalingPolicyConfiguration:
        PredefinedMetricSpecification:
          PredefinedMetricType: ElastiCachePrimaryEngineCPUUtilization
        TargetValue: 40
```

Escalado programado

El escalado según una programación le permite escalar la aplicación en respuesta a los cambios previstos en la demanda. Para utilizar el escalado programado, deberá crear acciones programadas, que indiquen a ElastiCache para Valkey y Redis OSS que realice actividades de escalado en momentos específicos. Cuando crea una acción programada, especifica un clúster existente, cuando debe ocurrir la actividad de escalado, la capacidad mínima y la capacidad máxima. Puede crear acciones programadas que realizan el escalado de forma puntual o periódica.

Solo puede crear una acción programada para los clústeres que ya existen. No puede crear una acción programada a la vez que crea un clúster.

A fin de obtener más información sobre la terminología de la creación, administración y eliminación de acciones programadas, consulte Comandos de uso frecuente para la creación, administración y eliminación de acciones programadas.

Para crear de forma periódica:

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Valkey o Redis OSS.
3. Elija el clúster para el que desea agregar una política.
4. Elija Manage Auto Scaling policies (Administrar política de Auto Scaling) en el menú desplegable de Actions (Acciones).
5. Elija la pestaña de Auto Scaling policies (Políticas de Auto Scaling).
6. En la sección de Auto Scaling policies (Políticas de Auto Scaling), aparece el cuadro de diálogo de Add Scaling policy (Agregar política de escalado). Elija Scheduled scaling (Escalado programado).
7. En Policy name (Nombre de política), ingrese el nombre de la política.

8. En Scalable Dimension (Dimensión escalable), elija Shards (Particiones).
9. En Target Shards (Particiones de destino), elija el valor.
10. En Recurrence (Recurrencia), elija Recurring (Recurrente).
11. En Frequency (Frecuencia), elija el valor respectivo.
12. En Start Date (Fecha de inicio) y Start time (Hora de inicio), elija la hora a partir de la cual la política entrará en vigor.
13. Elija Add Policy (Agregar política).

Para crear una acción programada puntual:

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Valkey o Redis OSS.
3. Elija el clúster para el que desea agregar una política.
4. Elija Manage Auto Scaling policies (Administrar política de Auto Scaling) en el menú desplegable de Actions (Acciones).
5. Elija la pestaña de Auto Scaling policies (Políticas de Auto Scaling).
6. En la sección de Auto Scaling policies (Políticas de Auto Scaling), aparece el cuadro de diálogo de Add Scaling policy (Agregar política de escalado). Elija Scheduled scaling (Escalado programado).
7. En Policy name (Nombre de política), ingrese el nombre de la política.
8. En Scalable Dimension (Dimensión escalable), elija Shards (Particiones).
9. En Target Shards (Particiones de destino), elija el valor.
10. En Recurrence (Recurrencia), elija One Time (Una vez).
11. En Start Date (Fecha de inicio) y Start time (Hora de inicio), elija la hora a partir de la cual la política entrará en vigor.
12. En End Date (Fecha de finalización), elija la fecha hasta la que la política estará vigente.
13. Elija Add Policy (Agregar política).

Para eliminar una acción programada

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.

2. En el panel de navegación, elija Valkey o Redis OSS.
3. Elija el clúster para el que desea agregar una política.
4. Elija Manage Auto Scaling policies (Administrar política de Auto Scaling) en el menú desplegable de Actions (Acciones).
5. Elija la pestaña de Auto Scaling policies (Políticas de Auto Scaling).
6. En la sección de Auto Scaling Policies (Políticas de Auto Scaling), elija la política de Auto Scaling y, a continuación, elija Delete (Eliminar) en el cuadro de diálogo de Actions (Acciones).

Para administrar el escalado programado mediante la AWS CLI

Utilice las siguientes API de escalado automático de aplicaciones:

- put-scheduled-action
- describe-scheduled-actions
- delete-scheduled-action

Utilice CloudFormation para crear una acción programada

Este fragmento muestra cómo crear una política de seguimiento de destino y aplicarla a un recurso AWS::ElastiCache::ReplicationGroup mediante el recurso AWS::ApplicationAutoScaling::ScalableTarget. Utiliza las funciones intrínsecas Fn::Join y Ref para construir la propiedad `ResourceId` con el nombre lógico del recurso `AWS::ElastiCache::ReplicationGroup` especificado en la misma plantilla.

```
ScalingTarget:
   Type: 'AWS::ApplicationAutoScaling::ScalableTarget'
   Properties:
     MaxCapacity: 3
     MinCapacity: 1
     ResourceId: !Sub replication-group/${logicalName}
     ScalableDimension: 'elasticache:replication-group:NodeGroups'
     ServiceNamespace: elasticache
     RoleARN: !Sub "arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/elasticache.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG"
     ScheduledActions:
       - EndTime: '2020-12-31T12:00:00.000Z'
```

```
        ScalableTargetAction:
          MaxCapacity: '5'
          MinCapacity: '2'
        ScheduledActionName: First
        Schedule: 'cron(0 18 * * ? *)'
```

Uso de Auto Scaling con réplicas

Un grupo de replicación de ElastiCache puede configurar una o más cachés para que funcionen como un solo nodo lógico.

A continuación se proporcionan detalles sobre el seguimiento de destinos y las políticas programadas, así como sobre cómo aplicarlas mediante la Consola de administración de AWS, la AWS CLI y API.

Políticas de escalado de seguimiento de destino

Las políticas de escalado de seguimiento de destino le permiten seleccionar una métrica y establecer un valor de destino. El escalado automático de ElastiCache para Valkey y Redis OSS crea y administra las alarmas de CloudWatch que desencadenan la política de escalado y calcula el ajuste de escalado en función de la métrica y el valor de destino. La política de escalado agrega o elimine las réplicas por igual en todas las particiones en función de las necesidades para mantener la métrica en el valor objetivo especificado o en un valor próximo. Además de mantener la métrica próxima al valor de destino, la política de escalado de seguimiento de destino también se ajusta a las fluctuaciones de la métrica producidas por patrones de carga fluctuante y minimiza las fluctuaciones rápidas de la capacidad de la flota.

Criterios de Auto Scaling para las réplicas

La política de Auto Scaling define la siguiente métrica predefinida para el clúster:

`ElastiCacheReplicaEngineCPUUtilization`: el umbral de utilización de AVG EngineCPU agregado en todas las réplicas que ElastiCache utiliza para activar una operación de escalado automático. Puede establecer el objetivo de utilización entre el 35 y el 70 %.

Cuando el servicio detecta que su métrica `ElastiCacheReplicaEngineCPUUtilization` es igual o mayor que la configuración del Target (Objetivo), aumentará las réplicas de las particiones de forma automática. ElastiCache escala horizontalmente las réplicas del clúster en un recuento igual al mayor de dos números: la variación del porcentaje del objetivo y una réplica. En el caso de la

reducción horizontal, ElastiCache no se reduce horizontalmente de forma automática a menos que el valor total de la métrica sea inferior al 75 % del objetivo definido.

Para un ejemplo de escalado horizontal, si tiene 5 particiones y 1 réplica cada una:

Si el objetivo se incumple en un 30 %, ElastiCache para Valkey y Redis OSS se escala horizontalmente por una réplica (máx. 0,3; el valor predeterminado es 1) en todas las particiones, lo que da como resultado cinco particiones con dos réplicas cada una.

Para ver un ejemplo de reducción horizontal, si ha seleccionado un valor de objetivo del 60 %, ElastiCache para Valkey y Redis OSS no se reducirá horizontalmente de forma automática hasta que la métrica sea inferior o igual al 45 % (25 % por debajo del objetivo del 60 %).

Consideraciones de Auto Scaling

Tenga en cuenta las siguientes consideraciones:

- En las políticas de escalado de seguimiento de destino, se presupone que el escalado ascendente se lleva a cabo cuando la métrica está por encima del valor de destino. No puede utilizar una política de escalado de seguimiento de destinos si la métrica especificada está por debajo del valor de destino. ElastiCache para Valkey y Redis OSS escala horizontalmente las réplicas con un máximo (porcentaje de desviación redondeada a partir del objetivo; el predeterminado es 1) de las réplicas existentes en todas las particiones del clúster.
- Las políticas de escalado de seguimiento de destino no realizan el escalado cuando la métrica especificada no tiene datos suficientes. No realiza el escalado porque la carencia de datos no se interpreta como una infrautilización de recursos.
- Es posible que haya diferencias entre el valor de destino y los puntos de datos de la métrica real. Esto se debe a que el escalado automático de ElastiCache siempre actúa de forma conservadora y redondea hacia arriba o hacia abajo a la hora de determinar la cantidad de capacidad que se debe añadir o quitar. Con esto se evita que se agrega capacidad insuficiente o se elimine demasiada capacidad.
- Para garantizar la disponibilidad de la aplicación, el servicio escalará horizontalmente en proporción a la métrica tan rápido como puede, pero se reduce horizontalmente de manera más gradual con un máximo de reducción horizontal de 1 réplica en las particiones del clúster.
- Puede tener varias políticas de escalado de seguimiento de destino en un clúster de ElastiCache para Valkey y Redis OSS, siempre que cada uno de ellos utilice una métrica diferente. El objetivo del escalado automático siempre es dar prioridad a la disponibilidad, por lo que su comportamiento varía en función de si las políticas de seguimiento de destino están listas para el escalado

horizontal o la reducción horizontal. Realizará un escalado horizontal del servicio si cualquiera de las políticas de seguimiento de destino está lista para el escalado horizontal, pero solo realizará la reducción horizontal si todas las políticas de seguimiento de destino (que tienen la parte de reducción horizontal habilitada) están listas para la reducción horizontal.

- No modifique ni elimine las alarmas de CloudWatch que administra el escalado automático de ElastiCache para las políticas de escalado de seguimiento de destino. El escalado automático elimina de forma automática las alarmas cuando se elimina la política de escalado o el clúster.
- El escalado automático de ElastiCache no impide modificar de forma manual las réplicas en las particiones. Estos ajustes manuales no afectan a las alarmas de CloudWatch existentes adjuntas a la política de escalado, pero pueden afectar a las métricas que pueden activar estas alarmas de CloudWatch.
- Estas alarmas de CloudWatch administradas por Auto Scaling se definen a través de la métrica AVG (Promedio) en todas las particiones del clúster. Por lo tanto, tener particiones activas puede resultar en cualquiera de los siguientes escenarios:
  - escalado cuando no es necesario debido a la carga en algunas particiones activas que desencadenan una alarma de CloudWatch
  - no escalar cuando sí sea necesario debido a un AVG (Promedio) agregado en todas las particiones que causa que la alarma se interrumpa.
- Siguen aplicándose los límites predeterminados de ElastiCache en nodos por clúster. Por lo tanto, al optar por Auto Scaling y, si espera que los nodos máximos sean superiores al límite predeterminado, solicite un aumento del límite en AWS Service Limits y elija el tipo de límite Nodes per cluster per instance type (Nodos por clúster por tipo de instancias).
- Asegúrese de tener suficientes ENI (interfaces de red elásticas) disponibles en su VPC, que son necesarias durante el escalado horizontal. Para obtener más información, consulte Interfaces de red elástica.
- Si no hay suficiente capacidad disponible de EC2, el escalado automático de ElastiCache no se escalará horizontalmente hasta que la capacidad se encuentre disponible o hasta que modifique de forma manual el clúster a los tipos de instancia que tengan suficiente capacidad.
- El escalado automático de ElastiCache no admite el escalado de réplicas con un clúster que tenga un `ReservedMemoryPercent` menor del 25 %. Para obtener más información, consulte Administración de la memoria reservada para Valkey y Redis OSS.

Adición de una política de escalado

Puede agregar una política de escalado mediante la Consola de administración de AWS.

Adición de una política de escalado mediante la Consola de administración de AWS

Cómo añadir una política de escalado automático a ElastiCache para Valkey y Redis OSS

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Valkey o Redis OSS.
3. Elija el clúster al que desea agregar una política (elija el nombre del clúster y no el botón a su izquierda).
4. Elija la pestaña de Auto Scaling policies (Políticas de Auto Scaling).
5. Elija add dynamic scaling (agregar escalado dinámico).
6. En Scaling policies (Políticas de escalado), elija Add dynamic scaling (Agregar escalado dinámico).
7. En Policy name (Nombre de política), ingrese el nombre de la política.
8. En Scalable Dimension (Dimensión escalable), seleccione Replicas (Réplicas) en el cuadro de diálogo.
9. Para el valor de objetivo, escriba el porcentaje promedio de utilización de la CPU que desea mantener en las réplicas de ElastiCache. El valor debe ser >=35 y <=70. Las réplicas de clúster se agregan o quitan para mantener la métrica en un valor próximo al especificado.
10. (Opcional) La consola no admite periodos de recuperación de reducción o escalado horizontal. Utilice la AWS CLI para modificar los valores de periodo de recuperación.
11. En Capacidad mínima, escriba el número mínimo de réplicas que debe mantener la política de escalado automático de ElastiCache.
12. En Capacidad máxima, escriba el número máximo de réplicas que debe mantener la política de escalado automático de ElastiCache. Este valor debe ser >=5.
13. Seleccione Crear.

Registro de un destino escalable

Puede aplicar una política de escalado en función de una métrica predefinida o una personalizada. Para ello, puede usar AWS CLI o la API de Auto Scaling de aplicaciones. El primer paso consiste en registrar su grupo de replicación de ElastiCache para Valkey y Redis OSS en el escalado automático.

Para poder utilizar el escalado automático de ElastiCache con un clúster, debe registrar su clúster en el escalado automático de ElastiCache. Esto se hace para definir la dimensión y los límites de

escalado que se van a aplicar a ese clúster. El escalado automático de ElastiCache escala de forma dinámica el clúster a lo largo de la dimensión escalable `elasticache:replication-group:Replicas`, que representa el número de réplicas del clúster por partición.

Uso de la CLI

Para registrar el clúster de ElastiCache, utilice el comando register-scalable-target con los siguientes parámetros:

- --service-namespace: establezca este valor en elasticache.
- --resource-id: el identificador de recurso para el clúster de ElastiCache. Para este parámetro, el tipo de recurso es ReplicationGroup y el identificador único es el nombre del clúster; por ejemplo, `replication-group/myscalablecluster`.
- --scalable-dimension: establezca este valor en `elasticache:replication-group:Replicas`.
- --min-capacity: número mínimo de réplicas que administrará el escalado automático de ElastiCache. Para obtener información sobre la relación entre --min-capacity, --max-capacity y el número de réplicas del clúster, consulte Capacidad mínima y máxima.
- --max-capacity: número máximo de réplicas que administrará el escalado automático de ElastiCache. Para obtener información sobre la relación entre --min-capacity, --max-capacity y el número de réplicas del clúster, consulte Capacidad mínima y máxima.

Example

En el siguiente ejemplo, registra un clúster de ElastiCache denominado `myscalablecluster`. El registro indica que el clúster debe escalarse de forma dinámica para tener de una a cinco réplicas.

Para Linux, macOS o Unix:

```
aws application-autoscaling register-scalable-target \
    --service-namespace elasticache \
    --resource-id replication-group/myscalablecluster \
    --scalable-dimension elasticache:replication-group:Replicas \
    --min-capacity 1 \
    --max-capacity 5 \
```

Para Windows:

```
aws application-autoscaling register-scalable-target ^
```

```
    --service-namespace elasticache ^
    --resource-id replication-group/myscalablecluster ^
    --scalable-dimension elasticache:replication-group:Replicas ^
    --min-capacity 1 ^
    --max-capacity 5 ^
```

Uso de la API

Para registrar el clúster de ElastiCache, utilice el comando register-scalable-target con los siguientes parámetros:

- ServiceNamespace: establezca este valor en elasticache.
- ResourceID: el identificador de recurso para el clúster de ElastiCache. Para este parámetro, el tipo de recurso es ReplicationGroup y el identificador único es el nombre del clúster; por ejemplo, `replication-group/myscalablecluster`.
- ScalableDimension: establezca este valor en `elasticache:replication-group:Replicas`.
- MinCapacity: número mínimo de réplicas que administrará el escalado automático de ElastiCache. Para obtener información sobre la relación entre --min-capacity, --max-capacity y el número de réplicas del clúster, consulte Capacidad mínima y máxima.
- MaxCapacity: número máximo de réplicas que administrará el escalado automático de ElastiCache. Para obtener información sobre la relación entre --min-capacity, --max-capacity y el número de réplicas del clúster, consulte Capacidad mínima y máxima.

Example

En el siguiente ejemplo, se registra un clúster denominado `myscalablecluster` con la API de Application Auto Scaling. Este registro indica que el clúster debe escalarse de forma dinámica para tener de una a cinco réplicas.

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.RegisterScalableTarget
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
```

```
{
    "ServiceNamespace": "elasticache",
    "ResourceId": "replication-group/myscalablecluster",
    "ScalableDimension": "elasticache:replication-group:Replicas",
    "MinCapacity": 1,
    "MaxCapacity": 5
}
```

Definición de una política de escalado

Una configuración de la política de escalado de seguimiento de destino está representada por un bloque JSON en el que se definen las métricas y los valores de destino. Puede guardar una configuración de la política de escalado como bloque JSON en un archivo de texto. Puede utilizar ese archivo de texto al invocar AWS CLI o la API de Auto Scaling de aplicaciones. Para obtener más información sobre la sintaxis de configuración de políticas, consulte TargetTrackingScalingPolicyConfiguration en la Referencia de la API de Auto Scaling de aplicaciones.

Las siguientes opciones se encuentran disponibles para definir una configuración de la política de escalado de seguimiento de destino:

Temas

- Uso de una métrica predefinida
- Edición de una política de escalado
- Eliminación de una política de escalado
- Uso de CloudFormation para las políticas de Auto Scaling
- Escalado programado

Uso de una métrica predefinida

Una configuración de la política de escalado de seguimiento de destino está representada por un bloque JSON en el que se definen las métricas y los valores de destino. Puede guardar una configuración de la política de escalado como bloque JSON en un archivo de texto. Puede utilizar ese archivo de texto al invocar AWS CLI o la API de Auto Scaling de aplicaciones. Para obtener más información sobre la sintaxis de configuración de políticas, consulte TargetTrackingScalingPolicyConfiguration en la Referencia de la API de Auto Scaling de aplicaciones.

Las siguientes opciones se encuentran disponibles para definir una configuración de la política de escalado de seguimiento de destino:

Temas

- Uso de una métrica predefinida
- Uso de una métrica personalizada
- Uso de periodos de recuperación
- Desactivación de actividad de escalado descendente
- Aplicación de una política de escalado a un clúster de ElastiCache para Valkey y Redis OSS

Uso de una métrica predefinida

Mediante las métricas predefinidas, puede definir rápidamente una política de escalado de seguimiento de destino para un clúster de ElastiCache para Valkey y Redis OSS que funciona con el seguimiento de destino en el escalado automático de ElastiCache. Actualmente, ElastiCache admite las siguientes métricas predefinidas en el escalado automático de réplicas de ElastiCache:

`ElastiCacheReplicaEngineCPUUtilization`: el valor medio de la métrica EngineCPUUtilization en CloudWatch en todas las réplicas del clúster. Puede encontrar el valor agregado de la métrica en CloudWatch en `ReplicationGroupId, Role` de ElastiCache para los valores de ReplicationGroupId y Role Replica necesarios.

Para usar una métrica predefinida en su política de escalado, puede crear una configuración de seguimiento de destino para su política de escalado. Esta configuración debe incluir `PredefinedMetricSpecification` para la métrica predefinida y `TargetValue` para el valor de destino de esa métrica.

Uso de una métrica personalizada

Mediante las métricas personalizadas, puede definir una política de escalado de seguimiento de destino que cumpla sus requisitos personalizados. Puede definir una métrica personalizada en función de cualquier métrica de ElastiCache para Valkey y Redis OSS que cambie en proporción al escalado. No todas las métricas de ElastiCache funcionan para el seguimiento de destino. La métrica debe ser una métrica de utilización válida y describir el nivel de actividad de una instancia. El valor de la métrica debe aumentar o reducirse en proporción al número de réplicas del clúster. Este aumento o reducción proporcionales son necesarios para utilizar los datos de las métricas a fin de ampliar o reducir proporcionalmente el número de réplicas.

Example

En el siguiente ejemplo se describe una configuración de seguimiento de destino para una política de escalado. En esta configuración, una métrica personalizada ajusta un clúster en función de un uso medio de la CPU del 50 % en todas las réplicas de un clúster denominado `my-db-cluster`.

```
{"TargetValue": 50,
    "CustomizedMetricSpecification":
    {"MetricName": "EngineCPUUtilization",
        "Namespace": "AWS/ElastiCache",
        "Dimensions": [
            {"Name": "ReplicationGroup","Value": "my-db-cluster"},
            {"Name": "Role","Value": "REPLICA"}
        ],
        "Statistic": "Average",
        "Unit": "Percent"
    }
}
```

Uso de periodos de recuperación

Puede especificar un valor, en segundos, a fin de que `ScaleOutCooldown` agregue un periodo de recuperación para el escalado horizontal de su clúster. De forma similar, puede agregar un valor, en segundos, a fin de que `ScaleInCooldown` agregue un periodo de recuperación para el escalado vertical de su clúster. Para obtener más información sobre `ScaleInCooldown` y `ScaleOutCooldown`, consulte TargetTrackingScalingPolicyConfiguration en la Referencia de la API de Auto Scaling de aplicaciones. En el siguiente ejemplo se describe una configuración de seguimiento de destino para una política de escalado. En esta configuración, la métrica predefinida `ElastiCacheReplicaEngineCPUUtilization` se usa para ajustar un clúster en función del uso medio de la CPU del 40 % en todas las réplicas de ese clúster. La configuración proporciona un periodo de recuperación de escalado descendente de 10 minutos y un periodo de recuperación de escalado ascendente de 5 minutos.

```
{"TargetValue": 40.0,
    "PredefinedMetricSpecification":
    {"PredefinedMetricType": "ElastiCacheReplicaEngineCPUUtilization"
    },
    "ScaleInCooldown": 600,
    "ScaleOutCooldown": 300
}
```

Desactivación de actividad de escalado descendente

Puede evitar que la configuración de la política de escalado de seguimiento de destino reduzca horizontalmente su clúster de ElastiCache para Valkey y Redis OSS desactivando la actividad de reducción horizontal. La deshabilitación de la actividad de escalado horizontal evita que la política de escalado elimine réplicas, a la vez que permite a la política de escalado agregarlas según sea necesario.

Puede especificar un valor booleano a fin de que `DisableScaleIn` habilite o deshabilite la actividad de reducción horizontal para su clúster. Para obtener más información sobre `DisableScaleIn`, consulte TargetTrackingScalingPolicyConfiguration en la Referencia de la API de Auto Scaling de aplicaciones.

Example

En el siguiente ejemplo se describe una configuración de seguimiento de destino para una política de escalado. En esta configuración, la métrica predefinida `ElastiCacheReplicaEngineCPUUtilization` ajusta un clúster en función del uso medio de la CPU del 40 % en todas las réplicas de ese clúster. La configuración deshabilita la actividad de escalado descendente para la política de escalado.

```
{"TargetValue": 40.0,
    "PredefinedMetricSpecification":
    {"PredefinedMetricType": "ElastiCacheReplicaEngineCPUUtilization"
    },
    "DisableScaleIn": true
}
```

Aplicación de una política de escalado a un clúster de ElastiCache para Valkey y Redis OSS

Tras registrar su clúster con escalado automático de ElastiCache para Valkey y Redis OSS y definir una política de escalado, puede aplicarla al clúster registrado. Para aplicar una política de escalado a un clúster de ElastiCache para Valkey y Redis OSS, puede utilizar la AWS CLI o la API de Escalado automático de aplicaciones.

Uso de AWS CLI

Para aplicar una política de escalado a un clúster de ElastiCache para Valkey y Redis OSS, utilice el comando put-scaling-policy con los siguientes parámetros:

- --policy-name: el nombre de la política de escalado.
- --policy-type: establezca este valor en `TargetTrackingScaling`.
- --resource-id: el identificador de recurso para el clúster. Para este parámetro, el tipo de recurso es ReplicationGroup y el identificador único es el nombre del clúster; por ejemplo, `replication-group/myscalablecluster`.
- --service-namespace: establezca este valor en elasticache.
- --scalable-dimension: establezca este valor en `elasticache:replication-group:Replicas`.
- --target-tracking-scaling-policy-configuration: la configuración de la política de escalado de seguimiento de destino que se utilizará para el clúster.

Example

En el siguiente ejemplo, se aplica una política de escalado de seguimiento de destino denominada `myscalablepolicy` a un clúster denominado `myscalablecluster` con escalado automático de ElastiCache. Para ello, puede usar una configuración de la política guardada en un archivo denominado `config.json`.

Para Linux, macOS o Unix:

```
aws application-autoscaling put-scaling-policy \
    --policy-name myscalablepolicy \
    --policy-type TargetTrackingScaling \
    --resource-id replication-group/myscalablecluster \
    --service-namespace elasticache \
    --scalable-dimension elasticache:replication-group:Replicas \
    --target-tracking-scaling-policy-configuration file://config.json
```

```
{"TargetValue": 40.0,
    "PredefinedMetricSpecification":
    {"PredefinedMetricType": "ElastiCacheReplicaEngineCPUUtilization"
    },
    "DisableScaleIn": true
}
```

Para Windows:

```
aws application-autoscaling put-scaling-policy ^
    --policy-name myscalablepolicy ^
    --policy-type TargetTrackingScaling ^
    --resource-id replication-group/myscalablecluster ^
    --service-namespace elasticache ^
    --scalable-dimension elasticache:replication-group:Replicas ^
    --target-tracking-scaling-policy-configuration file://config.json
```

Uso de la API

Para aplicar una política de escalado a un clúster de ElastiCache con la API de Escalado automático de aplicaciones, utilice la operación PutScalingPolicy de la API de Escalado automático de aplicaciones con los siguientes parámetros:

- PolicyName: el nombre de la política de escalado.
- PolicyType: establezca este valor en `TargetTrackingScaling`.
- ResourceID: el identificador de recurso para el clúster. Para este parámetro, el tipo de recurso es ReplicationGroup y el identificador único es el nombre del clúster de ElastiCache para Redis OSS, por ejemplo, `replication-group/myscalablecluster`.
- ServiceNamespace: establezca este valor en elasticache.
- ScalableDimension: establezca este valor en `elasticache:replication-group:Replicas`.
- TargetTrackingScalingPolicyConfiguration: la configuración de la política de escalado de seguimiento de destino que se utilizará para el clúster.

Example

En el siguiente ejemplo, se aplica una política de escalado de seguimiento de destino denominada `scalablepolicy` a un clúster denominado `myscalablecluster` con escalado automático de ElastiCache. Puede usar una configuración de la política en función de la métrica predefinida `ElastiCacheReplicaEngineCPUUtilization`.

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.PutScalingPolicy
X-Amz-Date: 20160506T182145Z
```

```
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
    "PolicyName": "myscalablepolicy",
    "ServiceNamespace": "elasticache",
    "ResourceId": "replication-group/myscalablecluster",
    "ScalableDimension": "elasticache:replication-group:Replicas",
    "PolicyType": "TargetTrackingScaling",
    "TargetTrackingScalingPolicyConfiguration": {
        "TargetValue": 40.0,
        "PredefinedMetricSpecification":
        {
            "PredefinedMetricType": "ElastiCacheReplicaEngineCPUUtilization"
        }
    }
}
```

Edición de una política de escalado

Puede editar una política de escalado utilizando la Consola de administración de AWS, la AWS CLI o la API de Auto Scaling de aplicaciones.

Edición de una política de escalado mediante la Consola de administración de AWS

Solo puede editar políticas con el tipo Predefined metrics (Métricas predefinidas) mediante la Consola de administración de AWS

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Valkey o Redis OSS.
3. Elija el clúster al que desea agregar una política (elija el nombre del clúster y no el botón a su izquierda).
4. Elija la pestaña de Auto Scaling policies (Políticas de Auto Scaling).
5. En Scaling policies (Políticas de escalado), elija el botón a la izquierda de la política de Auto Scaling que desea cambiar y, a continuación, elija Modify (Modificar).
6. Realice los cambios solicitados en la política.
7. Elija Modificar.
8. Realice cambios en la política.

9. Elija Modificar.

Edición de una política de escalado mediante la AWS CLI o la API de Auto Scaling de aplicaciones

Puede utilizar AWS CLI o la API de Application Auto Scaling para editar una política de escalado de la misma forma que aplica una política de escalado:

- Al usar la API de Auto Scaling de aplicaciones, especifique el nombre de la política que desea editar en el parámetro `PolicyName`. Especifique nuevos valores para los parámetros que desea cambiar.

Para obtener más información, consulte Aplicación de una política de escalado a un clúster de ElastiCache para Valkey y Redis OSS.

Eliminación de una política de escalado

Puede eliminar una política de escalado mediante la Consola de administración de AWS, la AWS CLI o la API de Auto Scaling de aplicaciones.

Eliminación de una política de escalado mediante la Consola de administración de AWS

Solo puede editar políticas con el tipo Predefined metrics (Métricas predefinidas) mediante la Consola de administración de AWS

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Valkey o Redis OSS.
3. Elija el clúster con la política de Auto Scaling que desea eliminar.
4. Elija la pestaña de Auto Scaling policies (Políticas de Auto Scaling).
5. En Scaling policies (Políticas de escalado), elija la política de Auto Scaling y, a continuación, elija Delete (Eliminar).

Eliminación de una política de escalado mediante la AWS CLI o la API de Auto Scaling de aplicaciones

Puede utilizar la AWS CLI o la API de Application Auto Scaling para eliminar una política de escalado de un clúster de ElastiCache.

CLI

Para eliminar una política de escalado de un clúster de ElastiCache para Valkey y Redis OSS, utilice el comando delete-scaling-policy con los siguientes parámetros:

- --policy-name: el nombre de la política de escalado.
- --resource-id: el identificador de recurso para el clúster. Para este parámetro, el tipo de recurso es ReplicationGroup y el identificador único es el nombre del clúster; por ejemplo, `replication-group/myscalablecluster`.
- --service-namespace: establezca este valor en elasticache.
- --scalable-dimension: establezca este valor en `elasticache:replication-group:Replicas`.

Example

En el siguiente ejemplo, elimina una política de escalado de seguimiento de destino denominada `myscalablepolicy` de un clúster de ELC denominado `myscalablecluster`.

Para Linux, macOS o Unix:

```
aws application-autoscaling delete-scaling-policy \
    --policy-name myscalablepolicy \
    --resource-id replication-group/myscalablecluster \
    --service-namespace elasticache \
    --scalable-dimension elasticache:replication-group:Replicas \
```

Para Windows:

```
aws application-autoscaling delete-scaling-policy ^
    --policy-name myscalablepolicy ^
    --resource-id replication-group/myscalablecluster ^
    --service-namespace elasticache ^
    --scalable-dimension elasticache:replication-group:Replicas ^
```

API

Para eliminar una política de escalado de su clúster de ElastiCache para Valkey y Redis OSS, utilice la operación DeleteScalingPolicy de la API de Escalado automático de aplicaciones con los siguientes parámetros:

- PolicyName: el nombre de la política de escalado.
- ResourceID: el identificador de recurso para el clúster. Para este parámetro, el tipo de recurso es ReplicationGroup y el identificador único es el nombre del clúster; por ejemplo, `replication-group/myscalablecluster`.
- ServiceNamespace: establezca este valor en elasticache.
- ScalableDimension: establezca este valor en `elasticache:replication-group:Replicas`.

En el siguiente ejemplo, elimina una política de escalado de seguimiento de destino denominada `myscalablepolicy` de un clúster llamado `myscalablecluster` con la API de Application Auto Scaling.

```
POST / HTTP/1.1
>>>>>>> mainline
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.DeleteScalingPolicy
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
    "PolicyName": "myscalablepolicy",
    "ServiceNamespace": "elasticache",
    "ResourceId": "replication-group/myscalablecluster",
    "ScalableDimension": "elasticache:replication-group:Replicas"
}
```

Uso de CloudFormation para las políticas de Auto Scaling

Este fragmento muestra cómo crear una acción programada y aplicarla a un recurso AWS::ElastiCache::ReplicationGroup mediante el recurso AWS::ApplicationAutoScaling::ScalableTarget. Utiliza las funciones intrínsecas Fn::Join y Ref para construir la propiedad `ResourceId` con el nombre lógico del recurso `AWS::ElastiCache::ReplicationGroup` especificado en la misma plantilla.

```
ScalingTarget:
   Type: 'AWS::ApplicationAutoScaling::ScalableTarget'
   Properties:
```

```
     MaxCapacity: 0
     MinCapacity: 0
     ResourceId: !Sub replication-group/${logicalName}
     ScalableDimension: 'elasticache:replication-group:Replicas'
     ServiceNamespace: elasticache
     RoleARN: !Sub "arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/elasticache.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG"

  ScalingPolicy:
    Type: "AWS::ApplicationAutoScaling::ScalingPolicy"
    Properties:
      ScalingTargetId: !Ref ScalingTarget
      ServiceNamespace: elasticache
      PolicyName: testpolicy
      PolicyType: TargetTrackingScaling
      ScalableDimension: 'elasticache:replication-group:Replicas'
      TargetTrackingScalingPolicyConfiguration:
        PredefinedMetricSpecification:
          PredefinedMetricType: ElastiCacheReplicaEngineCPUUtilization
        TargetValue: 40
```

Escalado programado

El escalado según una programación le permite escalar la aplicación en respuesta a los cambios previstos en la demanda. Para utilizar el escalado programado, deberá crear acciones programadas, que indiquen a ElastiCache para Valkey y Redis OSS que realice actividades de escalado en momentos específicos. Cuando se crea una acción programada, se especifica un clúster de ElastiCache existente, cuando se debe producir la actividad de escalado, la capacidad mínima y la capacidad máxima. Puede crear acciones programadas que realizan el escalado de forma puntual o periódica.

Solo puede crear una acción programada para los clústeres de ElastiCache que ya existen. No puede crear una acción programada a la vez que crea un clúster.

A fin de obtener más información sobre la terminología de la creación, administración y eliminación de acciones programadas, consulte Comandos de uso frecuente para la creación, administración y eliminación de acciones programadas.

Para crear una acción programada puntual:

Similar a la dimensión Shard (Partición). Consulte Escalado programado .

Para eliminar una acción programada

Similar a la dimensión Shard (Partición). Consulte Escalado programado .

Para administrar el escalado programado mediante la AWS CLI

Utilice las siguientes API de escalado automático de aplicaciones:

- put-scheduled-action
- describe-scheduled-actions
- delete-scheduled-action

Utilice CloudFormation para crear políticas de Auto Scaling

Este fragmento muestra cómo crear una acción programada y aplicarla a un recurso AWS::ElastiCache::ReplicationGroup mediante el recurso AWS::ApplicationAutoScaling::ScalableTarget. Utiliza las funciones intrínsecas Fn::Join y Ref para construir la propiedad `ResourceId` con el nombre lógico del recurso `AWS::ElastiCache::ReplicationGroup` especificado en la misma plantilla.

```
ScalingTarget:
   Type: 'AWS::ApplicationAutoScaling::ScalableTarget'
   Properties:
     MaxCapacity: 0
     MinCapacity: 0
     ResourceId: !Sub replication-group/${logicalName}
     ScalableDimension: 'elasticache:replication-group:Replicas'
     ServiceNamespace: elasticache
     RoleARN: !Sub "arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/elasticache.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG"
     ScheduledActions:
       - EndTime: '2020-12-31T12:00:00.000Z'
         ScalableTargetAction:
           MaxCapacity: '5'
           MinCapacity: '2'
         ScheduledActionName: First
         Schedule: 'cron(0 18 * * ? *)'
```

Modificación del modo de clúster

Valkey y Redis OSS son bases de datos en memoria distribuidas que admiten las funciones de partición y replicación. Los clústeres de ElastiCache Valkey y Redis OSS son la implementación distribuida que permite particionar los datos en varios nodos. Un clúster de ElastiCache para Redis OSS tiene dos modos de funcionamiento, el modo de clúster habilitado (CME) y el modo de clúster desactivado (CMD). En el modo CME, un motor de Valkey y Redis OSS funciona como una base de datos distribuida con múltiples particiones y nodos, mientras que en el modo CMD, Valkey y Redis OSS funcionan como un solo nodo.

Antes de migrar de CMD a CME, deben cumplirse las siguientes condiciones:

Important

La configuración del modo de clúster solo se puede cambiar del modo de clúster desactivado al modo de clúster habilitado. No es posible revertir esta configuración.

- Es posible que el clúster solo tenga claves en la base de datos 0.
- Las aplicaciones deben usar un cliente de Valkey o Redis OSS que sea capaz de usar el protocolo de clúster y un punto de conexión de configuración.
- La conmutación por error automática debe estar habilitada en el clúster con un mínimo de 1 réplica.
- La versión mínima del motor requerida para la migración es Valkey 7.2 y versiones posteriores o Redis OSS 7.0 y versiones posteriores.

Para migrar de CMD a CME, la configuración del modo de clúster solo se puede cambiar del modo de clúster desactivado al modo de clúster habilitado. Se trata de un procedimiento de dos pasos que garantiza la disponibilidad del clúster durante el proceso de migración.

Note

Debe proporcionar un grupo de parámetros con una configuración habilitada para el clúster, es decir, el parámetro habilitado para el clúster se establece como `yes`. Si utiliza un grupo de parámetros predeterminado, ElastiCache para Redis OSS seleccionará automáticamente el grupo de parámetros predeterminado correspondiente con una configuración habilitada para clústeres. El valor del parámetro habilitado para el clúster se establece en no para un clúster

de CMD. A medida que el clúster pasa al modo compatible, el valor del parámetro habilitado para el clúster se actualiza a `yes` como parte de la acción de modificación.
Para obtener más información, consulte Configuración de los parámetros de motor mediante grupos de parámetros de ElastiCache

1. Prepararse: cree un clúster de CME de prueba y asegúrese de que la pila esté lista para trabajar con él. ElastiCache para Redis OSS no tiene forma de verificar si está preparado. Para obtener más información, consulte Creación de un clúster para Valkey o Redis OSS.
2. Modificar la configuración del clúster de CMD existente para que sea compatible con el modo de clúster: en este modo, se implementará una única partición y ElastiCache para Redis OSS funcionará como un solo nodo, pero también como un solo clúster de partición. El modo compatible significa que la aplicación cliente puede usar cualquiera de los protocolos para comunicarse con el clúster. En este modo, las aplicaciones se deben volver a configurar para empezar a utilizar el protocolo del clúster de Valkey o Redis OSS y el punto de conexión de configuración. Para cambiar el modo de clúster de Valkey o Redis OSS al modo de clúster compatible, siga los pasos que se indican a continuación:

   > **Note**
   >
   > En el modo compatible, no se permiten otras operaciones de modificación, como el escalado y la versión del motor, para el clúster. Además, los parámetros (excluyendo `cacheParameterGroupName`) no se pueden modificar al definir el parámetro en modo clúster en la solicitud ModifyReplicationGroup.

   a. Con la Consola de administración de AWS, consulte Modificación de un grupo de reproducción y configure el modo de clúster en Compatible
   b. Mediante la API, consulte ModifyReplicationGroup y actualice el parámetro `ClusterMode` a `compatible`.
   c. Mediante la AWS CLI, consulte modify-replication-group y actualice el parámetro `cluster-mode` a `compatible`.

   Tras cambiar el modo de clúster de Valkey o Redis OSS al modo de clúster compatible, la API DescribeReplicationGroups devolverá el punto de conexión de configuración del clúster de ElastiCache para Redis OSS. El punto de conexión de configuración del clúster es un punto de

conexión único que las aplicaciones pueden utilizar para conectarse al clúster. Para obtener más información, consulte Búsqueda de puntos de conexión en ElastiCache.

3. Modificar la configuración del clúster al modo de clúster habilitado: una vez que el modo de clúster esté configurado como compatible con el modo de clúster, el segundo paso consiste en modificar la configuración del clúster para que esté habilitado el modo de clúster. En este modo, se ejecuta una única partición y los clientes ahora pueden escalar los clústeres o modificar otras configuraciones de clústeres.

   Para cambiar el modo de clúster a Habilitado, siga estos pasos:

   Antes de empezar, asegúrese de que los clientes de Valkey o Redis OSS hayan migrado al protocolo de clúster y de que el punto de conexión de la configuración del clúster no esté en uso.

   a. Con la Consola de administración de AWS, consulte Modificación de un grupo de reproducción y configure el modo de clúster en Habilitado.
   b. Mediante la API, consulte ModifyReplicationGroup y actualice el parámetro `ClusterMode` a `enabled`.
   c. Mediante la AWS CLI, consulte modify-replication-group y actualice el parámetro `cluster-mode` a `enabled`.

   Tras cambiar el modo de clúster a habilitado, los puntos de conexión se configurarán según la especificación del clúster de Valkey o Redis OSS. La API DescribeReplicationGroups devolverá el parámetro del modo de clúster como `enabled` y los puntos de conexión del clúster que ahora están disponibles para que las aplicaciones los utilicen para conectarse al clúster.

   Tenga en cuenta que los puntos de conexión del clúster cambiarán una vez que el modo de clúster se cambie a habilitado. Asegúrese de actualizar las aplicaciones con los nuevos puntos de conexión.

También puede elegir la opción de volver al modo de clúster desactivado (CMD) desde el modo de clúster compatible y conservar las configuraciones originales.

Modificar la configuración del clúster al modo de clúster desactivado desde el modo de clúster compatible

1. Con la Consola de administración de AWS, consulte Modificación de un grupo de reproducción y configure el modo de clúster en Deshabilitado

2. Mediante la API, consulte ModifyReplicationGroup y actualice el parámetro `ClusterMode` a `disabled`.
3. Mediante la AWS CLI, consulte modify-replication-group y actualice el parámetro `cluster-mode` a `disabled`.

Tras cambiar el modo de clúster a desactivado, la API DescribeReplicationGroups devolverá el parámetro de modo de clúster como `disabled`.

Replicación en regiones de AWS mediante los almacenes de datos globales

> **Note**
>
> Actualmente, el almacén de datos global solo está disponible para clústeres basados en nodos.

Al utilizar la característica Almacén de datos global, puede trabajar con una replicación de clústeres de Valkey o Redis OSS completamente administrada, rápida, fiable y segura en todas las regiones de AWS. Con esta característica, puede crear clústeres de réplicas de lectura entre regiones para permitir lecturas de baja latencia y recuperación ante desastres en todas las regiones de AWS.

En las secciones siguientes, puede encontrar una descripción de cómo trabajar con almacenes de datos globales.

Temas

- Descripción general
- Requisitos previos y limitaciones
- Uso de almacenes de datos globales (consola)
- Uso de almacenes de datos globales (CLI)

Descripción general

Cada almacén de datos global es un conjunto de uno o más clústeres que se replican entre sí.

Un almacén de datos global consta de lo siguiente:

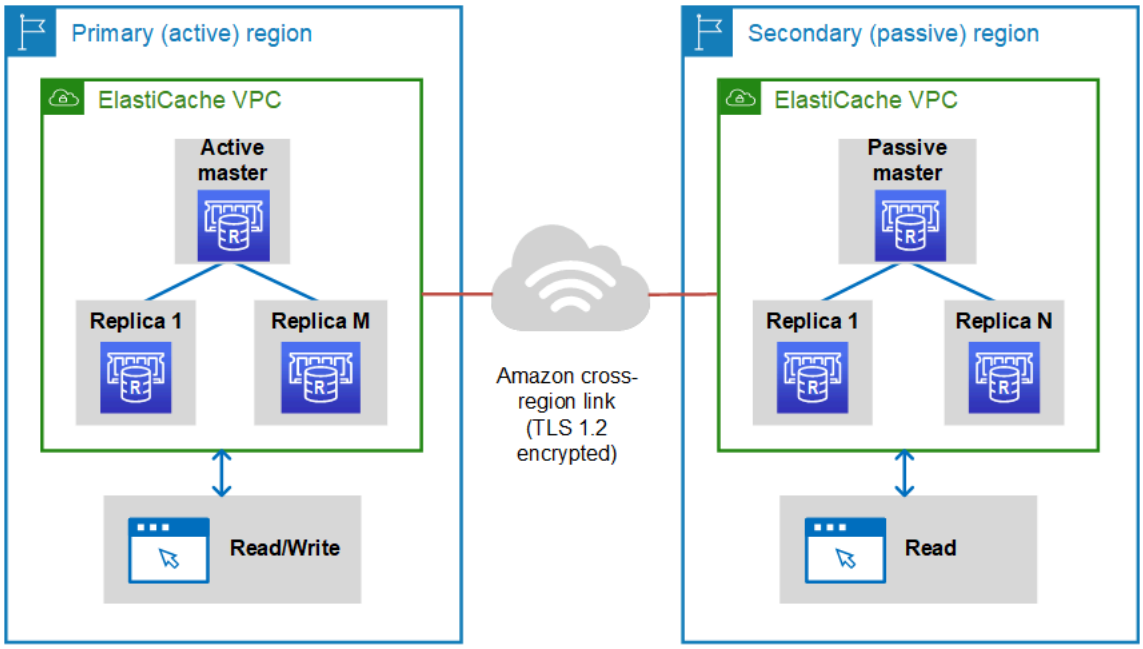
- Clúster principal (activo): un clúster principal acepta escrituras que se replican en todos los clústeres dentro del almacén de datos global. Un clúster principal también acepta solicitudes de lectura.
- Clúster secundario (pasivo): un clúster secundario solo acepta solicitudes de lectura y replica las actualizaciones de datos a partir de un clúster principal. Un clúster secundario debe encontrarse en una región de AWS diferente a la del principal.

Cuando crea un almacén de datos global en ElastiCache para Valkey o Redis OSS, replica de forma automática los datos del clúster principal al clúster secundario. Elija la región de AWS en la que se deben replicar los datos de Valkey o Redis OSS y, a continuación, cree un clúster secundario en esa región de AWS. ElastiCache configura y administra la reproducción automática y asincrónica de datos entre los dos clústeres.

El uso de un almacén de datos global de Valkey or Redis OSS ofrece las siguientes ventajas:

- Rendimiento geolocal: al configurar clústeres de réplicas remotas en regiones de AWS adicionales y sincronizar sus datos entre ellas, puede reducir la latencia del acceso a los datos en esa región de AWS. Un almacén de datos global puede contribuir a aumentar la capacidad de respuesta de su aplicación al ofrecer lecturas geolocales de baja latencia en todas las regiones de AWS.
- Recuperación de desastres: si el clúster principal de un almacén de datos global experimenta una reducción, puede promocionar un clúster secundario como su clúster principal nuevo. Puede hacerlo al conectarse a cualquier región de AWS que contenga un clúster secundario.

El siguiente diagrama muestra cómo pueden funcionar los almacenes de datos globales.



Requisitos previos y limitaciones

Antes de comenzar con almacenes de datos globales, tenga en cuenta lo siguiente:

- Los almacenes de datos globales se admiten en las siguientes regiones de AWS:
  - África: Ciudad del Cabo
  - Asia-Pacífico: Hong Kong, Hyderabad, Yakarta, Malasia, Melbourne, Mumbai, Osaka, Seúl, Singapur, Sídney, Tailandia y Tokio
  - Canadá: Canadá centro y Oeste de Canadá (Calgary)
  - China: Pekín y Ningxia
  - Europa: Fráncfort, Londres, Irlanda, Milán, París, España, Estocolmo y Zúrich
  - AWS GovCloud: Oeste de EE. UU. y Este de EE. UU.
  - Israel: Tel Aviv
  - Medio Oriente: Baréin y Emiratos Árabes Unidos
  - EE. UU.: Este (Norte de Virginia y Ohio) y Oeste (Norte de California y Oregón)
  - América del Sur: México (centro) y São Paulo
- Todos los clústeres (principal y secundario) del almacén de datos global deben tener el mismo número de nodos primarios, tipo de nodo, versión del motor y número de particiones (en caso de que el modo clúster se encuentre habilitado). Cada clúster del almacén de datos global puede tener un número distinto de réplicas de lectura para acomodar el tráfico de lectura local a dicho clúster.

La reproducción debe estar habilitada si tiene previsto utilizar un clúster de nodo único existente.

- Los almacenes de datos globales se admiten en instancias de tamaño grande y superior.
- Puede configurar la reproducción para un clúster principal desde una región de AWS a un clúster secundario en un máximo de otras dos regiones de AWS.

> **Note**
>
> La excepción a esto son las regiones de China (Pekín) y China (Ningxia), donde la reproducción solo puede ocurrir entre las dos regiones.

- Solo puede trabajar con almacenes de datos globales en clústeres de VPC. Para obtener más información, consulte Patrones de acceso para obtener acceso a una caché de ElastiCache en una Amazon VPC. Los almacenes de datos globales no se admiten cuando se utiliza EC2-Classic. Para obtener más información, consulte EC2-Classic en la Guía del usuario de Amazon EC2.

> **Note**
>
> Por el momento, no puede utilizar almacenes de datos globales en Uso de Local Zones con ElastiCache.

- ElastiCache no admite la conmutación por error automática de una región de AWS a otra. Cuando sea necesario, puede promover manualmente un clúster secundario. Para ver un ejemplo, consulta Promoción del clúster secundario a principal.
- Para iniciar desde datos existentes, utilice un clúster existente como principal para crear un almacén de datos global. No se admite la adición de un clúster existente como secundario. El proceso de agregar el clúster como secundario borra datos, lo que puede provocar la pérdida de datos.
- Las actualizaciones de parámetros se aplican a todos los clústeres cuando se modifica un grupo de parámetros local de un clúster que pertenece a un almacén de datos global.
- Puede escalar los clústeres regionales tanto vertical (aumento y disminución) como horizontalmente (escalado ascendente y descendente). Puede escalar los clústeres modificando el almacén de datos global. A continuación, todos los clústeres regionales del almacén de datos global se escalan sin interrupción. Para obtener más información, consulte Escalado de ElastiCache.
- Los almacenes de datos globales admiten cifrado en reposo, cifrado en tránsito y AUTH.

- Los almacenes de datos globales no admiten el protocolo de Internet versión 6 (IPv6).
- Los almacenes de datos globales admiten claves de AWS KMS. Para obtener más información, consulte Conceptos de servicios administrados de claves de AWS en la Guía para desarrolladores de AWS Key Management Service.

> **Note**
>
> Los almacenes de datos globales admiten mensajes de publicación/suscripción con las siguientes condiciones:
>
> - Para el modo de clúster deshabilitado, se admite por completo la publicación/suscripción. Los eventos publicados en el clúster principal de la región de AWS principal se propagan a las regiones de AWS secundarias.
> - Para el modo de clúster habilitado, se aplica lo siguiente:
>   - En el caso de los eventos publicados que no se encuentran en un espacio de claves, solo los suscriptores de la misma región de AWS reciben los eventos.
>   - En el caso de eventos de espacio de claves publicados, los suscriptores de todas las regiones de AWS reciben los eventos.

Uso de almacenes de datos globales (consola)

Para crear un almacén de datos global mediante la consola, siga este proceso de dos pasos:

1. Cree un clúster principal, ya sea mediante un clúster existente o creando un clúster nuevo. El motor debe ser Valkey 7.2 o una versión posterior, o Redis OSS 5.0.6 o una versión posterior.
2. Añada hasta dos clústeres secundarios en distintas regiones de AWS, utilizando de nuevo Valkey 7.2 o una versión posterior o Redis OSS 5.0.6 o una versión posterior.

Los siguientes procedimientos lo guían para crear un almacén de datos global para Valkey o Redis OSS y para realizar otras operaciones mediante la consola de ElastiCache.

Temas

- Creación de un almacén de datos global mediante un clúster existente
- Creación de un almacén de datos global nuevo mediante un clúster principal nuevo
- Visualización de detalles del almacén de datos global

- Adición de una región a un almacén de datos global
- Modificación de un almacén de datos global
- Promoción del clúster secundario a principal
- Eliminación de una región de un almacén de datos global
- Eliminación de un almacén de datos global

Creación de un almacén de datos global mediante un clúster existente

En esta situación, utilice un clúster existente que sirva como principal del nuevo almacén de datos global. A continuación, creará un clúster secundario de solo lectura en una región de AWS independiente. Este clúster secundario recibe actualizaciones automáticas y asincrónicas del clúster principal.

> **Important**
>
> El clúster existente debe utilizar un motor que sea Valkey 7.2 o posterior o Redis OSS 5.0.6 o posterior.

Para crear un almacén de datos global mediante un clúster existente

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Almacenes de datos globales y, luego, elija Crear almacén de datos global.
3. En la página Configuración del clúster principal, haga lo siguiente:
   - En el campo Información del almacén de datos global, introduzca un nombre para el nuevo almacén de datos global.
   - (Opcional) Introduzca un valor Description (Descripción).
4. En Clúster regional, seleccione Usar clúster regional existente.
5. En Clúster existente, seleccione el clúster existente que desea usar.
6. Mantenga las siguientes opciones tal como están. Se rellenan previamente para que coincidan con la configuración del clúster principal, no se pueden cambiar.
   - Versión del motor

- Tipo de nodo
- Grupo de parámetros

> **Note**
>
> ElastiCache genera de forma automática un grupo de parámetros nuevo a partir de los valores del grupo de parámetros proporcionado y aplica el grupo de parámetros nuevo al clúster. Utilice este nuevo grupo de parámetros para modificar parámetros en un almacén de datos global. Cada grupo de parámetros generado automáticamente está asociado a un solo clúster y, por lo tanto, solo a un almacén de datos global.

- Número de fragmentos
- Encryption at rest (Cifrado en reposo): permite el cifrado de los datos almacenados en el disco. Para obtener más información, consulte Cifrado en reposo.

> **Note**
>
> Puede suministrar una clave de cifrado distinta al elegir la Clave de AWS KMS administrada por el cliente y, a continuación elija la clave. Para obtener más información, consulte Uso de claves de AWS KMS administradas por el cliente.

- Encryption in-transit (Cifrado en tránsito): permite el cifrado de datos del cable. Para obtener más información, consulte Cifrado en tránsito. Para Valkey 7.2 en adelante y Redis OSS 6.0 en adelante, si habilita el cifrado en tránsito, se le pedirá que especifique una de las siguientes opciones de Control de acceso:
  - No Access Control (Sin control de acceso): este es el valor predeterminado. Esto indica que no hay restricciones.
  - Lista de control de acceso (ACL) de grupos de usuarios: elija un grupo de usuarios con un conjunto definido de usuarios y permisos para las operaciones disponibles. Para obtener más información, consulte Administración de grupos de usuarios con la consola y la CLI.
  - Usuario predeterminado con el comando AUTH: un mecanismo de autenticación para el servidor de Valkey o Redis OSS. Para obtener más información, consulte AUTH.

7. (Opcional) Según sea necesario, actualice la configuración del clúster secundario restante. Estos se rellenan previamente con los mismos valores que el clúster principal, pero puede actualizarlos para que cumplan los requisitos específicos de ese clúster.

- Puerto
- Número de réplicas
- Grupo de subredes
- Zonas de disponibilidad preferidas
- Grupos de seguridad
- Clave administrada por el cliente (clave de AWS KMS)
- Token de AUTH
- Habilitar copias de seguridad automáticas
- Backup retention period
- Intervalo de copia de seguridad
- Periodo de mantenimiento
- Tema para la notificación de SNS

8. Seleccione Crear. Al hacerlo, se establece el estado del almacén de datos global en Creating (Creando). El estado pasa a Modifying (Modificando) después de que el clúster principal esté asociado al almacén de datos global y el clúster secundario esté en estado Associating (Asociando).

   Una vez que el clúster principal y los clústeres secundarios se asocian al almacén de datos global, el estado cambia a Available (Disponible). En este punto, tiene un clúster principal que acepta lecturas y escrituras y clústeres secundarios que aceptan lecturas replicadas desde el clúster principal.

   La página se actualiza para indicar si un clúster forma parte de un almacén de datos global, lo que incluye:

   - Almacén de datos global: el nombre del almacén de datos global al que pertenece el clúster.
   - Rol del almacén de datos global: el rol del clúster, ya sea principal o secundario.

Puede agregar hasta un clúster secundario adicional en una región de AWS diferente. Para obtener más información, consulte Adición de una región a un almacén de datos global.

Creación de un almacén de datos global nuevo mediante un clúster principal nuevo

Si decide crear un almacén de datos global con un clúster nuevo, utilice el siguiente procedimiento.

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Almacenes de datos globales y, luego, elija Crear almacén de datos global.
3. En Primary cluster settings (Configuración de clúster principal), realice la siguiente operación:
    a. Para Cluster mode (Modo clúster), elija Enabled (Habilitado) o Disabled (Desactivado).
    b. Para Global Datastore info (Información del almacén de datos global) ingrese un valor para Name (Nombre). ElastiCache utiliza el sufijo para generar un nombre único para el almacén de datos global. Puede buscar el almacén de datos global utilizando el sufijo que especifique aquí.
    c. (Opcional) Escriba un valor para Global Datastore Description (Descripción de almacén de datos global).
4. En Regional cluster (Clúster regional):
    a. Para Region (Región), elija una región de AWS disponible.
    b. Elija Create new regional cluster (Crear un nuevo clúster regional) o Use existing regional cluster (Utilizar el clúster regional existente)
    c. Si elige Create new regional cluster (Crear un nuevo clúster regional), en Cluster info (Información sobre clúster), introduzca un nombre y una descripción opcional del clúster.
    d. En Location (Ubicación), le recomendamos que acepte la configuración predeterminada de Multi-AZ y Auto-failover (Conmutación por error automática).
5. En Cluster settings (Configuración del clúster)
    a. Para Engine version (Versión del motor), elija una versión disponible, que sea 5.0.6 o una versión posterior.
    b. En Port (Puerto), utilice el puerto predeterminado, 6379. Si tiene un motivo para utilizar un puerto diferente, escriba el número de puerto.
    c. En Grupo de parámetros, elija un grupo de parámetros o cree uno nuevo. Los grupos de parámetros controlan los parámetros de tiempo de ejecución de su clúster. Para obtener más información acerca de los grupos de parámetros, consulte Parámetros de Valkey y Redis OSS y Creación de un grupo de parámetros de ElastiCache.

> **Note**
>
> Cuando se selecciona un grupo de parámetros para definir los valores de configuración del motor, dicho grupo de parámetros se aplica a todos los clústeres del almacén de datos global. En la página Parameter Groups (Grupos de parámetros), el atributo Global sí/no, indica si un grupo de parámetros forma parte de un almacén de datos global.

d. Para Node type, elija la flecha hacia abajo (▼).
En el cuadro de diálogo Cambiar tipo de nodo, elija un valor para Familia de instancias para el tipo de nodo que desea. A continuación, elija el tipo de nodo que desea utilizar para este clúster y, a continuación, elija Guardar.

   Para obtener más información, consulte Elección del tamaño del nodo.

   Si elige un tipo de nodo r6gd, la organización de datos en niveles se activará automáticamente. Para obtener más información, consulte Organización de datos en niveles en ElastiCache.

e. Si está creando un clúster de Valkey o Redis OSS (modo de clúster deshabilitado):

   Para Number of replicas (Número de réplicas), elija el número de réplicas que desea para este clúster.

f. Si está creando un clúster de Valkey or Redis OSS (modo de clúster habilitado):

   i. En Número de particiones, elija el número de particiones (particiones/grupos de nodos) que desea para este clúster de Valkey o Redis OSS (modo de clúster habilitado).

      En algunas versiones de Valkey o Redis OSS (modo de clúster habilitado), puede cambiar dinámicamente el número de particiones del clúster:

      - Redis OSS 3.2.10 y versiones posteriores: si el clúster ejecuta Redis OSS 3.2.10 o versiones posteriores, puede cambiar el número de particiones del clúster de forma dinámica. Para obtener más información, consulte Escalado de clústeres de Valkey o Redis OSS (modo de clúster habilitado).
      - Otras versiones de Redis OSS: si su clúster ejecuta una versión de Redis OSS anterior a 3.2.10, existe otro enfoque. Para cambiar el número de particiones del

clúster en este caso, cree un nuevo clúster con el nuevo número de particiones. Para obtener más información, consulte Restauración desde una copia de seguridad a una nueva caché.

ii. En Réplicas por partición, elija el número de nodos de réplica de lectura que desea en cada partición.

Existen las siguientes restricciones para Valkey o Redis OSS (modo de clúster habilitado).

- Si tiene habilitado Multi-AZ, asegúrese de tener al menos una réplica por partición.
- El número de réplicas es el mismo para cada fragmento al crear el clúster utilizando la consola.
- El número de réplicas de lectura por fragmento es fijo y no puede modificarse. Si se da cuenta de que necesita más o menos réplicas por fragmento (API/CLI: grupo de nodos), debe crear un nuevo clúster con el nuevo número de réplicas. Para obtener más información, consulte Tutorial: propagación de un nuevo clúster basado en nodos con una copia de seguridad creada externamente.

6. Para Subnet group settings (Configuración de grupos de subredes), elija la subred que desea aplicar a este clúster. ElastiCache proporciona un grupo de subredes IPv4 predeterminado o puede optar por crear uno nuevo. Para IPv6, debe crear un grupo de subred con un bloque de CIDR IPv6. Si elige dual stack (doble pila), debe seleccionar un tipo de IP de detección, IPv6 o IPv4.

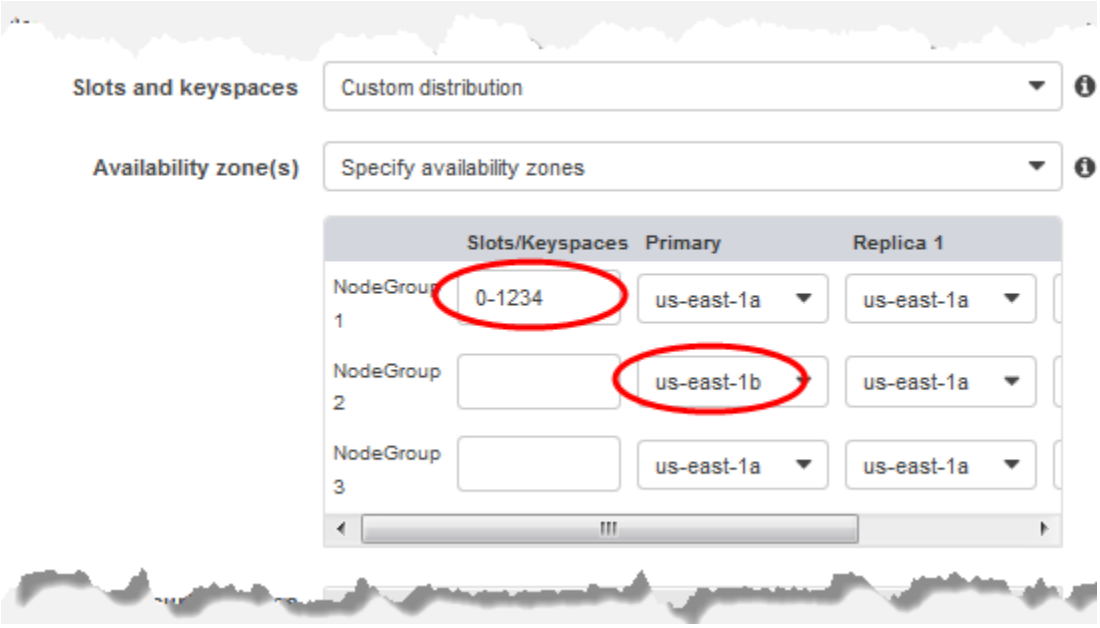
   Para obtener más información, consulte Creación de una subred en la VPC.

7. En Availability zone placements Ubicaciones de las zonas de disponibilidad), tiene dos opciones:

   - No preference (Sin preferencia): ElastiCache elige la zona de disponibilidad.
   - Specify availability zones (Especificar zonas de disponibilidad): especifique la zona de disponibilidad para cada clúster.

     Si decide especificar las zonas de disponibilidad, para cada clúster en cada fragmento, elija la zona de disponibilidad de la lista.

   Para obtener más información, consulte Elección de regiones y zonas de disponibilidad para ElastiCache.



Especificación de espacios de nombres y zonas de disponibilidad

8. Elija Next (Siguiente).
9. En Configuración avanzada de Valkey y Redis OSS

   - Para Security (Seguridad):

     i. Para cifrar sus datos, tiene las siguientes opciones:

     - Encryption at rest (Cifrado en reposo): permite el cifrado de los datos almacenados en el disco. Para obtener más información, consulte Cifrado en reposo.

       > **Note**
       >
       > Tiene la opción de suministrar una clave de cifrado distinta al elegir Customer Managed AWS KMS key (Clave KMS de administrada por el cliente) y elegir la clave. Para obtener más información, consulte Uso de claves de AWS KMS administradas por el cliente.

     - Encryption in-transit (Cifrado en tránsito): permite el cifrado de datos del cable. Para obtener más información, consulte Cifrado en tránsito. Para Valkey 7.2 y versiones posteriores o Redis OSS 6.0 y versiones posteriores, si habilita el cifrado en tránsito, se le pedirá que especifique una de las siguientes opciones de Control de acceso:

- No Access Control (Sin control de acceso): este es el valor predeterminado. Esto indica que no hay restricciones en el acceso de los usuarios al clúster.
- User Group Access Control List (Lista de control de acceso del grupo de usuarios): seleccione un grupo de usuarios con un conjunto definido de usuarios que pueden acceder al clúster. Para obtener más información, consulte Administración de grupos de usuarios con la consola y la CLI.
- Usuario predeterminado con el comando AUTH: un mecanismo de autenticación para el servidor de Valkey o Redis OSS. Para obtener más información, consulte AUTH.

- AUTH: un mecanismo de autenticación para un servidor de Valkey o Redis OSS. Para obtener más información, consulte AUTH.

> **Note**
>
> Para las versiones de Redis OSS a partir de la 3.2.6, excepto la versión 3.2.10, AUTH es la única opción.

ii. En Grupos de seguridad, elija los grupos de seguridad que desea para este clúster. Un grupo de seguridad actúa como un firewall para controlar el acceso de red al clúster. Puede utilizar el grupo de seguridad predeterminado para la VPC o crear uno nuevo.

Para obtener más información sobre los grupos de seguridad, consulte Grupos de seguridad de su VPC en la Guía del usuario de Amazon VPC.

10. En las copias de seguridad automáticas programadas regularmente, seleccione Enable automatic backups (Habilitar copias de seguridad automáticas) y, a continuación, escriba el número de días que desea conservar cada copia de seguridad automática antes de que se elimine automáticamente. Si no quiere tener copias de seguridad automáticas programadas regularmente, desactive la casilla de verificación Enable automatic backups (Habilitar copias de seguridad automáticas). En cualquier caso, siempre tendrá la opción de crear copias de seguridad manuales.

    Para obtener más información acerca de la copia de seguridad y restauración, consulte Instantánea y restauración.

11. (Opcional) Especifique un periodo de mantenimiento. El periodo de mantenimiento hace alusión al tiempo, generalmente una hora, de cada semana en la que ElastiCache programa el mantenimiento del sistema en su clúster. Puede permitir que ElastiCache elija el día y la

hora de su periodo de mantenimiento (No preference [Sin preferencia]), o bien puede elegir el día, la hora y la duración por su cuenta (Specify maintenance window [Especificar periodo de mantenimiento]). Si elige Especificar el periodo de mantenimiento en las listas, seleccione Día de inicio, Hora de inicio y Duración (en horas) para el periodo de mantenimiento. Todas las horas se indican en UCT.

Para obtener más información, consulte Administración del mantenimiento de clústeres de ElastiCache.

12. (Opcional) Para Logs (Registros):
    - En Log format (Formato de registro), elija Text o JSON.
    - En Destination Type (Tipo de destino), elija CloudWatch Logs o Kinesis Firehose.
    - En Destino de registro, elija Crear nuevo e introduzca el nombre del grupo de registro de Registros de CloudWatch o el nombre del flujo de Firehose, o elija Seleccionar existente y, a continuación, elija el nombre del grupo de registro de Registros de CloudWatch o el nombre del flujo de Firehose.
13. Para Tags (Etiquetas), para ayudarlo a administrar sus clústeres y otros recursos de ElastiCache, puede asignar sus propios metadatos a cada recurso en forma de etiquetas. Para obtener más información, consulte Etiquetado de los recursos de ElastiCache.
14. Revise todas las entradas y opciones y, a continuación, realice todos los cambios necesarios. Cuando esté preparado para continuar, seleccione Next (Siguiente).
15. Una vez que haya configurado el clúster en los pasos anteriores, configure los detalles de clúster secundario.
16. En Regional cluster (Clúster regional), elija la región de AWS en la que se encuentra el clúster.
17. En Cluster info (Información sobre clúster), introduzca un nombre y una descripción opcional del clúster.
18. Las siguientes opciones se rellenan previamente para que coincidan con la configuración del clúster principal y no se pueden cambiar:
    - Ubicación
    - Versión del motor
    - Tipo de instancia
    - Tipo de nodo
    - Número de fragmentos
    - Grupo de parámetros

> **Note**
>
> ElastiCache genera de forma automática un grupo de parámetros nuevo a partir de los valores del grupo de parámetros proporcionado y aplica el grupo de parámetros nuevo al clúster. Utilice este nuevo grupo de parámetros para modificar parámetros en un almacén de datos global. Cada grupo de parámetros generado automáticamente está asociado a un solo clúster y, por lo tanto, solo a un almacén de datos global.

- Encryption at rest (Cifrado en reposo): permite el cifrado de los datos almacenados en el disco. Para obtener más información, consulte Cifrado en reposo.

  > **Note**
  >
  > Puede suministrar una clave de cifrado distinta al elegir la Clave de AWS KMS administrada por el cliente y, a continuación elija la clave. Para obtener más información, consulte Uso de claves de AWS KMS administradas por el cliente.

- Encryption in-transit (Cifrado en tránsito): permite el cifrado de datos del cable. Para obtener más información, consulte Cifrado en tránsito. Para Valkey 7.2 y versiones posteriores o Redis OSS 6.4 y versiones posteriores, si habilita el cifrado en tránsito, se le pedirá que especifique una de las siguientes opciones de Control de acceso:
  - No Access Control (Sin control de acceso): este es el valor predeterminado. Esto indica que no hay restricciones en el acceso de los usuarios al clúster.
  - User Group Access Control List (Lista de control de acceso del grupo de usuarios): elija un grupo de usuarios con un conjunto definido de usuarios que pueden acceder al clúster. Para obtener más información, consulte Administración de grupos de usuarios con la consola y la CLI.
  - Usuario predeterminado con el comando AUTH: un mecanismo de autenticación para el servidor de Valkey o Redis OSS. Para obtener más información, consulte AUTH.

> **Note**
>
> Para las versiones de Redis OSS comprendidas entre la 4.0.2, cuando se admitió el cifrado en tránsito por primera vez, y la 6.0.4, AUTH es la única opción.

La configuración de clúster secundario restante se rellena previamente con los mismos valores que el clúster principal, pero se puede actualizar lo siguiente para cumplir los requisitos específicos de dicho clúster:

- Puerto
- Número de réplicas
- Grupo de subredes
- Zonas de disponibilidad preferidas
- Grupos de seguridad
- Clave administrada por el cliente (clave de AWS KMS)
- Token de AUTH
- Habilitar copias de seguridad automáticas
- Backup retention period
- Intervalo de copia de seguridad
- Periodo de mantenimiento
- Tema para la notificación de SNS

19. Seleccione Crear. Esto establece el estado del almacén de datos global en Creating (Creando). Una vez que el clúster principal y los clústeres secundarios se asocian al almacén de datos global, el estado cambia a Available (Disponible). Tiene un clúster principal que acepta lecturas y escrituras y un clúster secundario que acepta lecturas replicadas desde el clúster principal.

    La página también se actualiza para indicar si un clúster forma parte de un almacén de datos global, incluido lo siguiente:

    - Almacén de datos global: el nombre del almacén de datos global al que pertenece el clúster.
    - Rol del almacén de datos global: el rol del clúster, ya sea principal o secundario.

Puede agregar hasta un clúster secundario adicional en una región de AWS diferente. Para obtener más información, consulte Adición de una región a un almacén de datos global.

Visualización de detalles del almacén de datos global

Puede ver los detalles de los almacenes de datos globales existentes y modificarlos también en la página Almacén de datos global.

Para ver los detalles del almacén de datos global

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Almacén de datos global y, a continuación, elija un almacén de datos global disponible.

A continuación, puede examinar las siguientes propiedades globales del almacén de datos:

- Global Datastore Name (Nombre del almacén de datos global): el nombre del almacén de datos global
- Description (Descripción): una descripción del almacén de datos global
- Status (Estado): las opciones son:
  - Creación
  - Modificar
  - Disponible
  - Eliminación
  - Solo principal: este estado indica que el almacén de datos global contiene solo un clúster principal. Todos los clústeres secundarios se eliminan o no se crean correctamente.
- Cluster Mode (Modo de clúster): habilitado o deshabilitado
- Versión del motor: la versión del motor de Valkey o Redis OSS que ejecuta el almacén de datos global
- Instance Node Type (Tipo de nodo de instancia): el tipo de nodo utilizado para el almacén de datos global
- Encryption at-rest (Cifrado en reposo): habilitado o deshabilitado
- Encryption in-transit (Cifrado en tránsito): habilitado o deshabilitado
- AUTH: habilitado o deshabilitado

Puede realizar los siguientes cambios en el almacén de datos global:

- Adición de una región a un almacén de datos global
- Eliminación de una región de un almacén de datos global
- Promoción del clúster secundario a principal

- Modificación de un almacén de datos global

La página Global Datastore también muestra los clústeres individuales que componen el almacén de datos global y las siguientes propiedades para cada uno de ellos:

- Región: la región de AWS donde se almacena el clúster
- Role (Rol): primario o secundario
- Cluster name (Nombre del clúster): el nombre del clúster
- Status (Estado): las opciones son:
  - Associating (Asociando): el clúster está en proceso de asociarse al almacén de datos global
  - Associated (Asociado): el clúster está asociado al almacén de datos global
  - Disassociating (Desasociando): proceso de eliminación de un clúster secundario del almacén de datos global mediante el nombre del almacén de datos global. Después de esto, el clúster secundario ya no recibe actualizaciones del clúster principal, sino que permanece como clúster independiente en esa región de AWS.
  - Desasociado: el clúster secundario se ha eliminado del almacén de datos global y ahora es un clúster independiente en su región de AWS.
- Retraso en la réplica del almacén de datos global: muestra un valor por región de AWS secundaria en el almacén de datos global. Es el retraso entre el nodo primario de la región secundaria y el nodo primario de la región principal. Para Valkey o Redis OSS en modo de clúster habilitado, el retraso indica el retardo máximo entre las particiones.

Adición de una región a un almacén de datos global

Puede agregar hasta una región de AWS adicional a un almacén de datos global existente. En esta situación, crea un clúster de solo lectura en una región de AWS independiente que recibe actualizaciones automáticas y asíncronas del clúster principal.

Para agregar una región de AWS a un almacén de datos global

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Almacenes de datos globales y, a continuación, elija un almacén de datos global existente.
3. Elija Agregar clúster regional y luego la región de AWS donde residirá el clúster secundario.

4. En Información del clúster, escriba un valor en Nombre y, de manera opcional, en Descripción para el clúster.
5. Mantenga las siguientes opciones tal como están. Se rellenan previamente para que coincidan con la configuración del clúster principal y no se pueden cambiar.

   - Versión del motor
   - Tipo de instancia
   - Tipo de nodo
   - Número de fragmentos
   - Grupo de parámetros

   > **Note**
   >
   > ElastiCache genera de forma automática un grupo de parámetros nuevo a partir de los valores del grupo de parámetros proporcionado y aplica el grupo de parámetros nuevo al clúster. Utilice este nuevo grupo de parámetros para modificar parámetros en un almacén de datos global. Cada grupo de parámetros generado automáticamente está asociado a un solo clúster y, por lo tanto, solo a un almacén de datos global.

   - Cifrado en reposo

   > **Note**
   >
   > Puede suministrar una clave de cifrado distinta al elegir la Clave de AWS KMS administrada por el cliente y, a continuación elija la clave.

   - Cifrado en tránsito
   - AUTH
6. (Opcional) Actualice la configuración del clúster secundario restante. Esta se rellena previamente con los mismos valores que el clúster principal, pero puede actualizarlos para que cumplan los requisitos específicos de ese clúster:

   - Puerto
   - Número de réplicas
   - Grupo de subredes
   - Zonas de disponibilidad preferidas

- Grupos de seguridad
- Clave de AWS KMS administrada por el cliente
- Token de AUTH
- Habilitar copias de seguridad automáticas
- Backup retention period
- Intervalo de copia de seguridad
- Periodo de mantenimiento
- Tema para la notificación de SNS

7. Elija Agregar.

Modificación de un almacén de datos global

Puede modificar las propiedades de los clústeres regionales. Solo puede haber una operación de modificación en curso en un almacén de datos global, con la excepción de la promoción de un clúster secundario a principal. Para obtener más información, consulte Promoción del clúster secundario a principal.

Para modificar un almacén de datos global

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Almacenes de datos globales y, a continuación, en Nombre del almacén de datos global, elija un almacén de datos global.
3. Elija Modify (Modificar) y elija entre las siguientes opciones:

   - Modificar descripción: actualice la descripción del almacén de datos global
   - Modificar versión del motor: solo están disponibles Valkey 7.2 y versiones posteriores o Redis OSS 5.0.6 y versiones posteriores.
   - Modificar tipo de nodo: escale los clústeres regionales tanto de forma vertical (reducción y escalado vertical) como horizontal (escalado y reducción horizontal). Las opciones incluyen las familias de nodos R5 y M5. Para obtener más información acerca de los tipos de nodos, consulte Tipos de nodos compatibles.
   - Modificar la conmutación por error automática: habilite o deshabilite la conmutación por error automática. Cuando habilita la conmutación por error y los nodos primarios de los clústeres regionales se desconectan de forma inesperada, ElastiCache conmuta por error a una de

las réplicas regionales. Para obtener más información, consulte Conmutación por error automática.

Para clústeres de Valkey o Redis OSS con el modo de clúster habilitado:

- Agregar particiones: ingrese el número de particiones que desea agregar y, opcionalmente, especifique una o varias zonas de disponibilidad.
- Eliminar particiones: elija particiones que se eliminarán en cada región de AWS.
- Particiones de reequilibrio: reequilibre la distribución de las ranuras para garantizar una distribución uniforme entre las particiones existentes en el clúster.

A fin de modificar los parámetros de un almacén de datos global, modifique el grupo de parámetros de cualquier clúster de miembros para el almacén de datos global. ElastiCache aplica este cambio a todos los clústeres de ese almacén de datos global de forma automática. Para modificar el grupo de parámetros de ese clúster, utilice la consola de Valkey o Redis OSS o la operación de la API ModifyCacheCluster. Para obtener más información, consulte Modificación de un grupo de parámetros de ElastiCache. Cuando modifica el grupo de parámetros de cualquier clúster contenido en un almacén de datos global, se aplica a todos los clústeres dentro de ese almacén de datos global.

Para restablecer un grupo de parámetros completo o parámetros específicos, utilice la operación de la API ResetCacheParameterGroup.

Promoción del clúster secundario a principal

Si el clúster principal o la región de AWS no se encuentran disponibles o tiene problemas de rendimiento, puede promover un clúster secundario a principal. La promoción está permitida en cualquier momento, incluso si hay otras modificaciones en curso. También puede emitir varias promociones en paralelo y el almacén de datos global se resuelve al final en uno principal. Si promociona varios clústeres secundarios de forma simultánea, ElastiCache no garantiza cuál se resuelve en última instancia en el principal.

Para promover un clúster secundario a principal

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Almacenes de datos globales.

3. Elija el nombre del almacén de datos global para ver los detalles.
4. Elija el clúster Secondary (Secundario).
5. Elija Promote to primary (Promocionar a principal).

   A continuación, se pedirá que confirme su decisión con la siguiente advertencia: `Promoting a region to primary will make the cluster in this region as read/writable. Are you sure you want to promote the secondary cluster to primary?`

   `The current primary cluster in primary region will become secondary and will stop accepting writes after this operation completes. Please ensure you update your application stack to direct traffic to the new primary region.`
6. Seleccione Confirm (Confirmar) si desea continuar la promoción o Cancel (Cancelar) si no lo desea.

Si opta por confirmar, el almacén de datos global pasará a un estado Modifying (Modificando) y no estará disponible hasta que se complete la promoción.

Eliminación de una región de un almacén de datos global

Puede quitar una región de AWS de un almacén de datos global mediante el siguiente procedimiento.

Para eliminar una región de AWS de un almacén de datos global

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Almacenes de datos globales.
3. Elija un almacén de datos global.
4. Seleccione la Region (Región) que desea eliminar.
5. Elija Remove region (Quitar región).

> **Note**
>
> Esta opción solo está disponible para clústeres secundarios.

A continuación, se le pedirá que confirme su decisión con la siguiente advertencia: `Removing the region will remove your only available cross region replica for the primary cluster. Your primary cluster will no longer be set up for disaster recovery and improved read latency in remote region. Are you sure you want to remove the selected region from the global datastore?`

6. Seleccione Confirm (Confirmar) si desea continuar la promoción o Cancel (Cancelar) si no lo desea.

Si elige confirmar, se eliminará la región de AWS y el clúster secundario dejará de recibir actualizaciones de reproducción.

Eliminación de un almacén de datos global

Para eliminar un almacén de datos global, quite primero todos los clústeres secundarios. Para obtener más información, consulte Eliminación de una región de un almacén de datos global. Al hacerlo, el almacén de datos global se encuentra en estado primary-only (solo principal).

Para eliminar un almacén de datos global

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Almacenes de datos globales.
3. En Global Datastore Name (Nombre del almacén de datos global), elija el almacén de datos que desea eliminar y, a continuación, elija Delete (Eliminar).

   A continuación, se le pedirá que confirme su decisión con la siguiente advertencia: `Are you sure you want to delete this Global Datastore?`
4. Elija Eliminar.

El almacén de datos global pasa al estado Deleting (Eliminando).

Uso de almacenes de datos globales (CLI)

Puede usar la AWS Command Line Interface (AWS CLI) para controlar varios servicios de AWS desde la línea de comandos y automatizarlos mediante scripts. Puede utilizar la AWS CLI para operaciones ad hoc (únicas).

Descarga y configuración de la AWS CLI

La AWS CLI se ejecuta en Windows, macOS o Linux. Utilice el siguiente procedimiento para descargarlo y configurarlo.

Para descargar, instalar y configurar la CLI

1. Descargue la CLI de AWS en la página web de Interfaz de línea de comandos de AWS.
2. Siga las instrucciones para instalar la AWS CLI y configurar la AWS CLI en la Guía del usuario de la AWS Command Line Interface.

Uso de la AWS CLI con almacenes de datos globales

Utilice las siguientes operaciones de la CLI para trabajar con almacenes de datos globales:

- create-global-replication-group

```
aws elasticache create-global-replication-group \
   --global-replication-group-id-suffix my global datastore  \
   --primary-replication-group-id sample-repl-group  \
   --global-replication-group-description an optional description of the global
 datastore
```

Amazon ElastiCache aplica de forma automática un prefijo al ID de almacén de datos global en el momento de la creación. Cada región de AWS tiene su propio prefijo. Por ejemplo, un ID de almacén de datos global creado en la región de oeste de EE. UU. (Norte de California) comienza con “virxk” junto con el nombre del sufijo que proporcione. El sufijo, en conjunto con el prefijo autogenerado, garantiza la exclusividad del nombre del almacén de datos global en varias regiones.

En la siguiente tabla se enumera cada región de AWS y su prefijo de ID de almacén de datos global.

Nombre de la región/Región	Prefix
Región del Este de EE. UU. (Ohio) `us-east-2`	`fpkhr`

Nombre de la región/Región	Prefix
Región del Este de EE. UU (Norte de Virginia) `us-east-1`	`ldgnf`
Región del Oeste de EE. UU (Norte de California) `us-west-1`	`virxk`
Región del Oeste de EE. UU (Oregón) `us-west-2`	`sgaui`
Región de Canadá (centro) `ca-central-1`	`bxodz`
Región de Asia-Pacífico (Bombay) `ap-south-1`	`erpgt`
Asia Pacífico (Tokio) `ap-northeast-1`	`quwsw`
Región de Asia-Pacífico (Seúl) `ap-northeast-2`	`lfqnh`
Región Asia-Pacífico (Osaka) `ap-northeast-3`	`nlapn`
Región de Asia-Pacífico (Singapur) `ap-southeast-1`	`vlqxn`
Región de Asia-Pacífico (Sídney) `ap-southeast-2`	`vbgxd`

Nombre de la región/Región	Prefix
Región de Europa (Fráncfort) `eu-central-1`	`iudkw`
Región de Europa (Irlanda) `eu-west-1`	`gxeiz`
Región de Europa (Londres) `eu-west-2`	`okuqm`
Región EU (París) `eu-west-3`	`fgjhi`
Región de América del Sur (São Paulo) `sa-east-1`	`juxlw`
Región China (Pekín) `cn-north-1`	`emvgo`
Región China (Ningxia) `cn-northwest-1`	`ckbem`
Región de Asia-Pacífico (Hong Kong) `ap-east-1`	`knjmp`
AWS GovCloud (Oeste de EE. UU.) `us-gov-west-1`	`sgwui`

- create-replication-group: utilice esta operación con el fin de crear clústeres secundarios para un almacén de datos global al proporcionar el nombre del almacén de datos global al parámetro `--global-replication-group-id`.

```
aws elasticache create-replication-group \
  --replication-group-id secondary replication group name \
  --replication-group-description “Replication group description" \
  --global-replication-group-id global datastore name
```

Al llamar a esta operación y pasar un valor de `--global-replication-group-id`, ElastiCache deducirá los valores del grupo de replicación principal del grupo de replicación global para los siguientes parámetros. No ingrese valores para estos parámetros:

`"PrimaryClusterId",`

`"AutomaticFailoverEnabled",`

`"NumNodeGroups",`

`"CacheParameterGroupName",`

`"CacheNodeType",`

`"Engine",`

`"EngineVersion",`

`"CacheSecurityGroupNames",`

`"EnableTransitEncryption",`

`"AtRestEncryptionEnabled",`

`"SnapshotArns",`

`"SnapshotName"`

- describe-global-replication-groups

```
aws elasticache describe-global-replication-groups \
   --global-replication-group-id my global datastore  \
   --show-member-info an optional parameter that returns a list of the primary and
 secondary clusters that make up the global datastore
```

- modify-global-replication-group

```
aws elasticache modify-global-replication-group \
   --global-replication-group-id my global datastore  \
   --automatic-failover-enabled \
   --cache-node-type node type \
   --cache-parameter-group-name parameter group name \
   --engine-version engine version \
   --–apply-immediately \
   --global-replication-group-description description
```

Actualización entre motores de Redis OSS a Valkey para GlobalDataStore de ElastiCache

Puede actualizar un grupo de replicación global de Redis OSS existente en Valkey mediante la consola, la API o la CLI.

Si ya tiene un grupo de replicación global de Redis OSS, puede actualizarlo a Valkey especificando el nuevo motor y la versión del motor con la API modify-global-replication-group.

Para Linux, macOS o Unix:

```
aws elasticache modify-global-replication-group \
   --global-replication-group-id myGlobalReplGroup \
   --engine valkey \
   --apply-immediately \
   --engine-version 8.0
```

Para Windows:

```
aws elasticache modify-global-replication-group ^
   --global-replication-group-id myGlobalReplGroup ^
   --engine valkey ^
   --apply-immediately ^
   --engine-version 8.0
```

Si tiene un grupo de parámetros de caché personalizado aplicado al grupo de replicación global de Redis OSS existente que desea actualizar, también tendrá que incluir un grupo de parámetros de caché de Valkey personalizado en la solicitud. El grupo de parámetros personalizados de Valkey de entrada debe tener los mismos valores de parámetros estáticos de Redis OSS que el grupo de parámetros personalizados de Redis OSS existente.

Para Linux, macOS o Unix:

```
aws elasticache modify-global-replication-group \
   --global-replication-group-id myGlobalReplGroup \
   --engine valkey \
   --engine-version 8.0 \
   --apply-immediately \
   --cache-parameter-group-name myParamGroup
```

Para Windows:

```
aws elasticache modify-global-replication-group ^
   --global-replication-group-id myGlobalReplGroup ^
   --engine valkey ^
   --engine-version 8.0 ^
   --apply-immediately ^
   --cache-parameter-group-name myParamGroup
```

- delete-global-replication-group

```
aws elasticache delete-global-replication-group \
   --global-replication-group-id my global datastore  \
   --retain-primary-replication-group defaults to true
```

- disassociate-global-replication-group

```
aws elasticache disassociate-global-replication-group \
   --global-replication-group-id my global datastore  \
   --replication-group-id my secondary cluster  \
   --replication-group-region the AWS Region in which the secondary cluster resides
```

- failover-global-replication-group

```
aws elasticache failover-replication-group \
   --global-replication-group-id my global datastore \
   --primary-region The AWS Region of the primary cluster \
   --primary-replication-group-id  The name of the global datastore, including the
 suffix.
```

- increase-node-groups-in-global-replication-group

```
aws elasticache increase-node-groups-in-global-replication-group \
   --apply-immediately yes \
   --global-replication-group-id global-replication-group-name \
   --node-group-count 3
```

- decrease-node-groups-in-global-replication-group

```
aws elasticache decrease-node-groups-in-global-replication-group \
   --apply-immediately yes \
   --global-replication-group-id global-replication-group-name \
   --node-group-count 3
```

- rebalance-shards-in-global-replication-group

```
aws elasticache rebalance-shards-in-global-replication-group \
   --apply-immediately yes \
   --global-replication-group-id global-replication-group-name
```

Utilice la ayuda para mostrar todos los comandos disponibles de ElastiCache para Valkey o Redis OSS.

```
aws elasticache help
```

También puede utilizar la ayuda para describir un comando específico y obtener más información sobre su uso:

```
aws elasticache create-global-replication-group help
```

Alta disponibilidad a través de grupos de reproducción

Los clústeres de Valkey o Redis OSS de Amazon ElastiCache de un único nodo son entidades en memoria con servicios de protección de datos limitados (AOF). Si el clúster produce un error por cualquier motivo, se perderán todos los datos del clúster. No obstante, si ejecuta el motor de Valkey o Redis OSS, puede agrupar de 2 a 6 nodos en un clúster con réplicas, donde de 1 a 5 nodos de solo lectura contienen datos de réplica del nodo principal de lectura/escritura único del grupo. En esta situación, si solo se produce un error en un nodo por cualquier motivo, no perderá todos los datos, ya

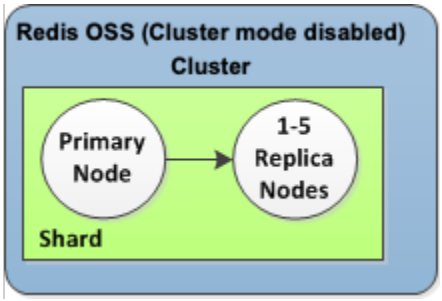
que estos se replican en otro u otros nodos. Debido a la latencia de reproducción, es posible que se pierdan algunos datos si el error se produce en el nodo principal de lectura/escritura.

Como se observa en el siguiente gráfico, la estructura de la replicación se encuentra en una partición (denominada grupo de nodos en la API/CLI), que a su vez se encuentra en un clúster de Valkey o Redis OSS. Los clústeres de Valkey o Redis OSS (modo de clúster deshabilitado) siempre tienen una partición. Los clústeres de Valkey o Redis OSS (modo de clúster habilitado) pueden tener hasta 500 particiones, con sus datos del clúster divididos entre las particiones. Puede crear un clúster con un mayor número de particiones y un menor número de réplicas con un total de hasta 90 nodos por clúster. Esta configuración de clúster puede variar desde 90 particiones y 0 réplicas hasta 15 particiones y 5 réplicas, que es el número máximo de réplicas permitido.

El límite de nodos o particiones se puede aumentar a un máximo de 500 por clúster con ElastiCache para Valkey o con ElastiCache versión 5.0.6 o posteriores para Redis OSS. Por ejemplo, puede elegir configurar un clúster de 500 nodos que oscila entre 83 particiones (uno primario y 5 réplicas por partición) y 500 particiones (único primario y sin réplicas). Asegúrese de que hay suficientes direcciones IP disponibles para acomodar el aumento. Algunos problemas comunes incluyen que las subredes del grupo de subredes tienen un rango CIDR demasiado pequeño o que otros clústeres comparten y utilizan considerablemente las subredes. Para obtener más información, consulte Creación de un grupo de subredes.

Para las versiones inferiores a 5.0.6, el límite es de 250 por clúster.

Para solicitar un aumento del límite, consulte AWS Service Limits y elija el tipo de límite Nodes per cluster per instance type (Nodos por clúster por tipo de instancias).



El clúster de Valkey o Redis OSS (modo de clúster deshabilitado) tiene una sola partición y de 0 a 5 nodos de réplica

Si el clúster con réplicas tiene habilitado Multi-AZ y el nodo principal produce un error, el principal realizará una conmutación por error a una réplica de lectura. Dado que los datos se actualizan en los nodos de réplica de forma asíncrona, es posible que haya algunas pérdidas de datos debido

a la latencia en la actualización de los nodos de réplica. Para obtener más información, consulte Mitigación de errores al ejecutar Valkey o Redis OSS.

Temas

- Descripción de la replicación de Valkey y Redis OSS
- Replicación: Valkey y Redis OSS en modo de clúster deshabilitado o habilitado
- Minimización del tiempo de inactividad en ElastiCache utilizando multi-AZ con Valkey y Redis OSS
- Cómo se implementan la sincronización y la copia de seguridad
- Creación de un grupo de replicación de Valkey o Redis OSS
- Visualización de detalles de un grupo de reproducción
- Búsqueda de puntos de enlace del grupo de reproducción
- Modificación de un grupo de reproducción
- Eliminación de un grupo de reproducción
- Cambio del número de réplicas
- Promoción de una réplica de lectura a principal para grupos de replicación de Valkey o Redis OSS (modo de clúster deshabilitado)

Descripción de la replicación de Valkey y Redis OSS

Redis OSS implementa la replicación de dos formas:

- Con una única partición que contiene todos los datos del clúster en cada nodo: Valkey o Redis OSS (modo de clúster deshabilitado)
- Con datos particionados en hasta 500 particiones: Valkey o Redis OSS (modo de clúster habilitado)

Cada partición de un grupo de reproducción tiene un nodo primario de lectura/escritura y hasta 5 nodos de réplica de solo lectura. Puede crear un clúster con un mayor número de particiones y un menor número de réplicas con un total de hasta 90 nodos por clúster. Esta configuración de clúster puede variar desde 90 particiones y 0 réplicas hasta 15 particiones y 5 réplicas, que es el número máximo de réplicas permitido.

El límite de nodos o particiones se puede aumentar a un máximo de 500 por clúster si la versión del motor de Redis OSS es 5.0.6 o posteriores. Por ejemplo, puede elegir configurar un clúster de 500 nodos que oscila entre 83 particiones (uno primario y 5 réplicas por partición) y 500 particiones (único primario y sin réplicas). Asegúrese de que hay suficientes direcciones IP disponibles para acomodar el aumento. Algunos problemas comunes incluyen que las subredes del grupo de subredes tienen un rango CIDR demasiado pequeño o que otros clústeres comparten y utilizan considerablemente las subredes. Para obtener más información, consulte Creación de un grupo de subredes.

Para las versiones inferiores a 5.0.6, el límite es de 250 por clúster.

Para solicitar un aumento del límite, consulte AWS Service Limits y elija el tipo de límite Nodes per cluster per instance type (Nodos por clúster por tipo de instancias).

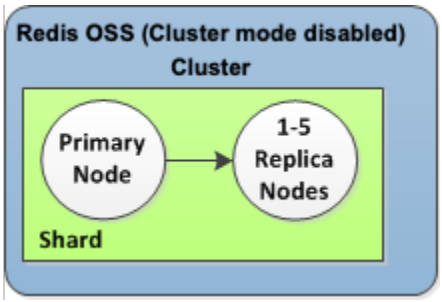
Temas

- Valkey o Redis OSS (modo de clúster deshabilitado)
- Valkey o Redis OSS (modo de clúster habilitado)

Valkey o Redis OSS (modo de clúster deshabilitado)

Los clústeres de Valkey o Redis OSS (modo de clúster deshabilitado) contienen una sola partición, que alberga una colección de nodos: un nodo principal de lectura/escritura y hasta cinco nodos de réplica secundarios de solo lectura. Cada réplica de lectura mantiene una copia de los datos del

nodo principal del clúster. Se utilizan mecanismos de replicación asíncronos para mantener las réplicas de lectura sincronizadas con el principal. Las aplicaciones pueden leer de cualquier nodos del clúster. Las aplicaciones pueden escribir únicamente en el nodo principal. Las réplicas de lectura mejoran el rendimiento de lectura y evitan la pérdida de datos en caso de error en un nodo.



Clúster de Valkey o Redis OSS (modo de clúster deshabilitado) con una sola partición y nodos de réplica

Puede utilizar clústeres de Valkey o Redis OSS (modo de clúster deshabilitado) con nodos de réplica para escalar su solución de ElastiCache para gestionar las aplicaciones con gran intensidad de lecturas o para permitir que un gran número de clientes pueda leer simultáneamente desde el mismo clúster.

Todos los nodos de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) deben residir en la misma región.

Al agregar una réplica de lectura a un clúster, todos los datos de la instancia principal se copian al nuevo nodo. A partir de entonces, siempre que se escriben datos en el nodo principal, los cambios se propagarán de forma asíncrona a todas las réplicas de lectura.

Para mejorar la tolerancia a errores y reducir el tiempo de inactividad de escritura, habilite multi-AZ con conmutación por error automática para el clúster de Valkey o Redis OSS (modo de clúster deshabilitado) con réplicas. Para obtener más información, consulte Minimización del tiempo de inactividad en ElastiCache utilizando multi-AZ con Valkey y Redis OSS.

Puede cambiar los roles de los nodos del clúster de Valkey o Redis OSS (modo de clúster deshabilitado), donde el nodo principal y una de las réplicas intercambian los roles. Puede que decida hacerlo por motivos de ajuste del rendimiento. Por ejemplo, con una aplicación web que tenga una gran actividad de escritura, puede elegir el nodo con la menor latencia de red. Para obtener más información, consulte Promoción de una réplica de lectura a principal para grupos de replicación de Valkey o Redis OSS (modo de clúster deshabilitado).

Valkey o Redis OSS (modo de clúster habilitado)

Un clúster de Valkey o Redis OSS (modo de clúster habilitado) se compone de 1 a 500 particiones (API/CLI: grupos de nodos). Cada partición tiene un nodo primario y hasta 5 nodos de réplica de solo lectura. La configuración puede variar desde 90 particiones y 0 réplicas hasta 15 particiones y 5 réplicas, que es el número máximo de réplicas permitido.

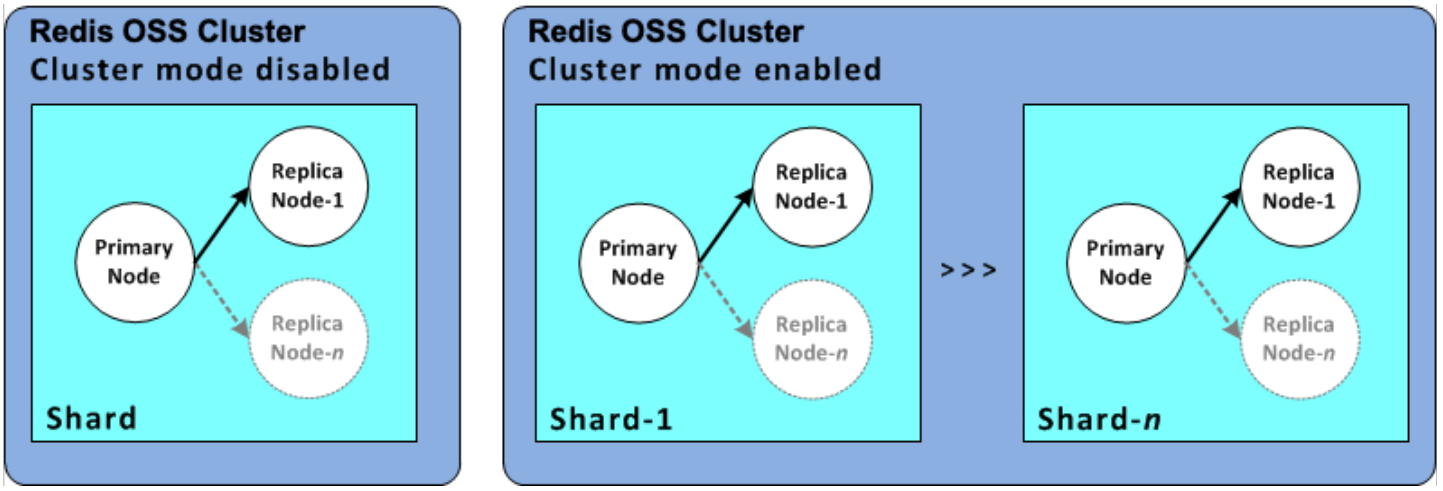
El límite de nodos o particiones se puede aumentar a un máximo de 500 por clúster si la versión del motor es Valkey 7.2 y posteriores o Redis OSS 5.0.6 y posteriores. Por ejemplo, puede elegir configurar un clúster de 500 nodos que oscila entre 83 particiones (uno primario y 5 réplicas por partición) y 500 particiones (único primario y sin réplicas). Asegúrese de que hay suficientes direcciones IP disponibles para acomodar el aumento. Algunos problemas comunes incluyen que las subredes del grupo de subredes tienen un rango CIDR demasiado pequeño o que otros clústeres comparten y utilizan considerablemente las subredes. Para obtener más información, consulte Creación de un grupo de subredes.

Para las versiones inferiores a 5.0.6, el límite es de 250 por clúster.

Para solicitar un aumento del límite, consulte AWS Service Limits y elija el tipo de límite Nodes per cluster per instance type (Nodos por clúster por tipo de instancias).

Cada réplica de lectura de un fragmento mantiene una copia de los datos del principal del fragmento. Se utilizan mecanismos de replicación asíncronos para mantener las réplicas de lectura sincronizadas con el principal. Las aplicaciones pueden leer de cualquier nodos del clúster. Las aplicaciones pueden escribir únicamente en los nodos. Las réplicas de lectura mejoran la escalabilidad de lectura y evitan la pérdida de datos. Los datos se encuentran divididos entre las distintas particiones del clúster de Valkey o Redis OSS (modo de clúster habilitado).

Las aplicaciones utilizan el punto de conexión de configuración del clúster de Valkey o Redis OSS (modo de clúster habilitado) para conectarse a los nodos del clúster. Para obtener más información, consulte Búsqueda de puntos de conexión en ElastiCache.



Clúster de Valkey o Redis OSS (modo de clúster habilitado) con varias particiones y nodos de réplica

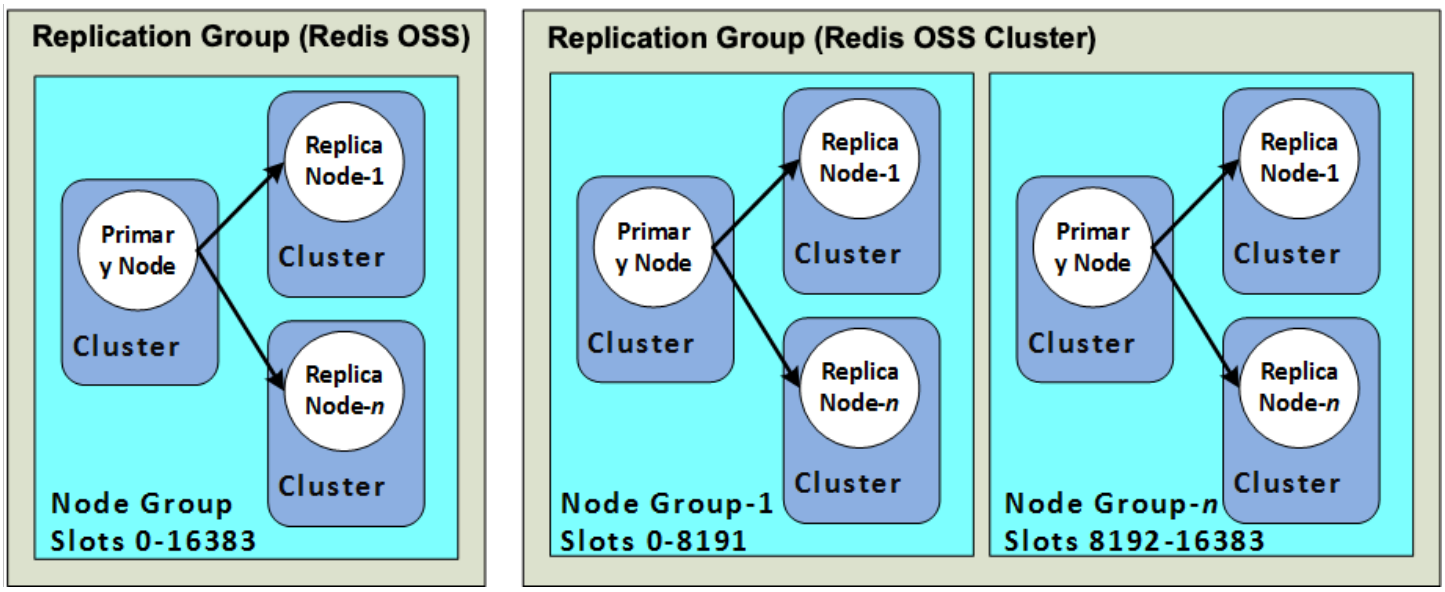
Todos los nodos de un clúster de Valkey o Redis OSS (modo de clúster habilitado) deben residir en la misma región. Para mejorar la tolerancia a errores, puede aprovisionar tanto a los principales como a las réplicas de lectura en varias zonas de disponibilidad dentro de esa región.

En la actualidad, las características de Valkey o Redis OSS (modo de clúster habilitado) tienen algunas limitaciones.

- No puede promocionar manualmente ningún nodo de réplica a principal.

Replicación: Valkey y Redis OSS en modo de clúster deshabilitado o habilitado

A partir de Valkey 7.2 y Redis OSS versión 3.2, tiene la posibilidad de crear uno de dos tipos distintos de clústeres (API/CLI: grupos de replicación). Un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) siempre dispone de una sola partición (API/CLI: grupo de nodos) con hasta 5 nodos de réplica de lectura. Un clúster de Valkey o Redis OSS (modo de clúster habilitado) tiene hasta 500 particiones con entre 1 y 5 nodos de réplica de lectura en cada uno.



Clústeres de Valkey o Redis OSS (modo de clúster deshabilitado) y Valkey o Redis OSS (modo de clúster habilitado)

En la siguiente tabla, se resumen las diferencias importantes entre los clústeres de Valkey o Redis OSS (modo de clúster deshabilitado) y Valkey o Redis OSS (modo de clúster habilitado).

Comparación de clústeres de Valkey o Redis OSS (modo de clúster deshabilitado) y Valkey o Redis OSS (modo de clúster habilitado)

Característica	Valkey o Redis OSS (modo de clúster deshabilitado)	Valkey o Redis OSS (modo de clúster habilitado)
Modificable	Sí. Admite la adición y eliminación de nodos de réplica, así como la ampliación del tipo de nodo.	Limitado. Para obtener más información, consulte Administración de versiones para ElastiCache y Escalado de clústeres de Valkey o Redis OSS (modo de clúster habilitado).
Particiones de datos	No	Sí
Particiones	1	1 a 500
Réplicas de lectura	De 0 a 5	0 a 5 por partición.

Característica	Valkey o Redis OSS (modo de clúster deshabilitado)	Valkey o Redis OSS (modo de clúster habilitado)
	Important Si no tiene réplicas y el nodo falla, experimentará la pérdida total de datos.	**Important** Si no tiene réplicas y el nodo falla, experimentará la pérdida de todos los datos del fragmento.
Multi-AZ	Sí, con al menos 1 réplica. Opcional. Activado de forma predeterminada.	Sí Opcional. Activado de forma predeterminada.
Instantáneas (copias de seguridad)	Sí, creando un único archivo .rdb.	Sí, creando un único archivo .rdb para cada fragmento.
Restaurar	Sí, al utilizar un único archivo .rdb de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado).	Sí, al utilizar archivos .rdb desde un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) o un clúster de Valkey o Redis OSS (modo de clúster habilitado).
Compatible con	Todas las versiones de Valkey y Redis OSS	Todas las versiones de Valkey y Redis OSS 3.2 y posteriores
Motor actualizable	Sí, con algunos límites. Para obtener más información, consulte Administración de versiones para ElastiCache.	Sí, con algunos límites. Para obtener más información, consulte Administración de versiones para ElastiCache.

Característica	Valkey o Redis OSS (modo de clúster deshabilitado)	Valkey o Redis OSS (modo de clúster habilitado)
Cifrado	Versiones 3.2.6 (programadas para EOL, consulte Programa de fin de vida útil de las versiones de Redis OSS), 4.0.10 y posteriores.	Versiones 3.2.6 (programadas para EOL, consulte Programa de fin de vida útil de las versiones de Redis OSS), 4.0.10 y posteriores.
Cumplimiento de requisitos de HIPAA	Versiones 3.2.6 (programadas para EOL, consulte Programa de fin de vida útil de las versiones de Redis OSS), 4.0.10 y posteriores.	Versiones 3.2.6 (programadas para EOL, consulte Programa de fin de vida útil de las versiones de Redis OSS), 4.0.10 y posteriores.
Compatible con PCI/DSS	Versiones 3.2.6 (programadas para EOL, consulte Programa de fin de vida útil de las versiones de Redis OSS), 4.0.10 y posteriores.	Versiones 3.2.6 (programadas para EOL, consulte Programa de fin de vida útil de las versiones de Redis OSS), 4.0.10 y posteriores.
Cambios de los fragmentos online	N/A	Versión 3.2.10 (programada para EOL, consulte Programa de fin de vida útil de las versiones de Redis OSS) y posteriores.

¿Cuál debería elegir?

Al elegir entre Valkey o Redis OSS (modo de clúster deshabilitado) o Valkey o Redis OSS (modo de clúster habilitado), tenga en cuenta los siguientes factores:

- Escalado frente a partición: las necesidades empresariales cambian. Debe aprovisionar los picos de demanda o escalar conforme cambie la demanda. Valkey o Redis OSS (modo de clúster deshabilitado) admite el escalado. Puede escalar la capacidad de lectura añadiendo o eliminando nodos de réplica, o bien puede escalar la capacidad ampliando a un tipo de nodo más grande.

Ambas operaciones requieren tiempo. Para obtener más información, consulte Escalado de nodos de réplica para Valkey o Redis OSS (modo de clúster deshabilitado).

Valkey o Redis OSS (modo de clúster habilitado) admite la partición de los datos en hasta 500 grupos de nodos. Puede cambiar el número de fragmentos de forma dinámica a medida que cambien sus necesidades de negocio. Una ventaja de la partición es que puede repartir la carga entre un mayor número de puntos de enlace, lo que reduce los cuellos de botella de acceso durante los picos de demanda. Además, puede alojar un mayor conjunto de datos, ya que los datos se pueden distribuir en varios servidores. Para obtener más información acerca del escalado de las particiones, consulte Escalado de clústeres de Valkey o Redis OSS (modo de clúster habilitado).

- Tamaño de nodo frente a número de nodos: puesto que un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) tiene una única partición, el tipo de nodo debe ser lo suficientemente grande como para alojar todos los datos del clúster, además de la capacidad adicional necesaria. Por otra parte, ya que puede particionar los datos entre varias particiones al utilizar un clúster de Valkey o Redis OSS (modo de clúster habilitado), los tipos de nodos pueden ser inferiores, aunque necesitará más cantidad. Para obtener más información, consulte Elección del tamaño del nodo.

- Lecturas frente a escrituras: si la carga principal en el clúster son los datos de lectura de las aplicaciones, puede escalar un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) añadiendo o eliminando réplicas de lectura. Sin embargo, hay un máximo de 5 réplicas de lectura. Si la carga de su clúster tiene un volumen de operaciones de escritura elevado, puede beneficiarse de los puntos de conexión de escritura adicionales de un clúster de Valkey o Redis OSS (modo de clúster habilitado) con varias particiones.

Independientemente del tipo de clúster que elija para implementar, debe asegurarse de elegir un tipo de nodo que se ajuste a sus necesidades actuales y futuras.

Minimización del tiempo de inactividad en ElastiCache utilizando multi-AZ con Valkey y Redis OSS

Existen varias instancias en las que ElastiCache para Valkey y Redis OSS podrían tener que reemplazar un nodo principal. Entre estas se incluyen determinados tipos de mantenimiento planificado y el caso poco probable de que se produzca un error en el nodo principal o en la zona de disponibilidad.

Este reemplazo produce un tiempo de inactividad para el clúster, pero si Multi-AZ se encuentra habilitado, el tiempo de inactividad es mínimo. El rol del nodo primario tendrá una conmutación por error automática en una de las réplicas de lectura. No es necesario crear ni aprovisionar un nodo primario nuevo, ya que ElastiCache se encargará de esto de forma clara. Esta conmutación por error y promoción de réplica garantizan la posibilidad de reanudar la escritura en la réplica principal tan pronto como se complete la promoción.

ElastiCache también propaga el nombre del servicio de nombres de dominio (DNS) de la réplica promocionada. Lo hace así porque, en ese caso, si su aplicación escribe en el punto de enlace principal, no se requiere ningún cambio de punto de conexión en su aplicación. Si lee desde puntos de conexión individuales, asegúrese de cambiar el punto de enlace de lectura de la réplica promovida a principal en el punto de enlace de la nueva réplica.

En caso de que se inicien reemplazos de nodos planificados debido a actualizaciones de mantenimiento o actualizaciones de autoservicio, tenga en cuenta lo siguiente:

- En el caso de clústeres de Valkey y Redis OSS, los reemplazos de nodos planificados se realizan mientras el clúster atiende las solicitudes de escritura que se reciben.
- En los clústeres de Valkey y Redis OSS que tienen el modo de clúster deshabilitado con multi-AZ habilitado y que se ejecutan en un motor de la versión 5.0.6 o posteriores, los reemplazos de nodos planificados se realizan mientras el clúster atiende las solicitudes de escritura que se reciben.
- En los clústeres de Valkey y Redis OSS que tienen el modo de clúster deshabilitado con multi-AZ habilitado y que se ejecutan en un motor de la versión 4.0.10 o anterior, es posible que se produzca una breve interrupción de escritura asociada con las actualizaciones de DNS. Esta interrupción es posible que tarde unos segundos. Este proceso es mucho más rápido que el de volver a crear y aprovisionar una réplica principal nueva, que es el proceso que se realiza en caso de no habilitar Multi-AZ.

Puede habilitar Multi-AZ mediante la consola de administración de ElastiCache, la AWS CLI o la API de ElastiCache.

Habilitar multi-AZ de ElastiCache en su clúster de Valkey o Redis OSS (grupo de replicación en la API o la CLI) mejora la tolerancia a errores. Esto es cierto especialmente en los casos en que el nodo principal de lectura/escritura del clúster deja de estar accesible o de funcionar por cualquier motivo. Multi-AZ solo se admite en clústeres de Valkey o Redis OSS que tienen más de un nodo en cada partición.

Temas

- Habilitación de Multi-AZ
- Escenarios de error con respuestas de Multi-AZ
- Prueba de la conmutación por error automática
- Limitaciones en multi-AZ

Habilitación de Multi-AZ

Puede habilitar Multi-AZ al crear o modificar un clúster (grupo de reproducción en la API o la CLI) mediante la consola de ElastiCache, la AWS CLI o la API de ElastiCache.

Solo puede habilitar multi-AZ en clústeres de Valkey o Redis OSS (modo de clúster deshabilitado) que tengan al menos una réplica de lectura disponible. Los clústeres sin réplicas de lectura no ofrecen alta disponibilidad ni tolerancia a errores. Para obtener información acerca de la creación de clústeres con reproducción, consulte Creación de un grupo de replicación de Valkey o Redis OSS. Para obtener información acerca de la adición de réplicas de lectura a un clúster con reproducción, consulte Adición de una réplica de lectura para Valkey o Redis OSS (modo de clúster deshabilitado).

Temas

- Habilitación de Multi-AZ (consola)
- Habilitación de Multi-AZ (AWS CLI)
- Habilitación de Multi-AZ (API de ElastiCache)

Habilitación de Multi-AZ (consola)

Puede habilitar multi-AZ mediante la consola de ElastiCache al crear un clúster de Valkey o Redis OSS nuevo o modificar un clúster existente con la replicación.

Multi-AZ se encuentra habilitado de forma predeterminada en los clústeres de Valkey o Redis OSS (modo de clúster habilitado).

> **Important**
>
> ElastiCache habilitará de forma automática Multi-AZ solo si el clúster contiene al menos una réplica en una zona de disponibilidad diferente de la principal en todas las particiones.

Habilitación de Multi-AZ al crear un clúster mediante la consola de ElastiCache

Para obtener más información acerca de este proceso, consulte Creación de un clúster de Valkey (modo de clúster deshabilitado) (consola). Asegúrese de tener una o más réplicas y habilitar Multi-AZ.

Habilitación de Multi-AZ en un clúster existente (consola)

Para obtener más información sobre este proceso, consulte Modificación de un clúster Uso de la Consola de administración de AWS de ElastiCache.

Habilitación de Multi-AZ (AWS CLI)

En el ejemplo de código siguiente se utiliza la AWS CLI para habilitar Multi-AZ para el grupo de reproducción `redis12`.

> **Important**
>
> El grupo de reproducción `redis12` debe existir y tener al menos una réplica de lectura disponible.

Para Linux, macOS o Unix:

```
aws elasticache modify-replication-group \
    --replication-group-id redis12 \
    --automatic-failover-enabled \
    --multi-az-enabled \
    --apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group ^
    --replication-group-id redis12 ^
    --automatic-failover-enabled ^
    --multi-az-enabled ^
    --apply-immediately
```

La salida JSON de este comando debería tener un aspecto similar al siguiente.

```
{
    "ReplicationGroup": {
        "Status": "modifying",
        "Description": "One shard, two nodes",
        "NodeGroups": [
            {
                "Status": "modifying",
                "NodeGroupMembers": [
                    {
                        "CurrentRole": "primary",
                        "PreferredAvailabilityZone": "us-west-2b",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Port": 6379,
                            "Address":
 "redis12-001.v5r9dc.0001.usw2.cache.amazonaws.com"
                        },
                        "CacheClusterId": "redis12-001"
                    },
                    {
                        "CurrentRole": "replica",
                        "PreferredAvailabilityZone": "us-west-2a",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Port": 6379,
                            "Address":
 "redis12-002.v5r9dc.0001.usw2.cache.amazonaws.com"
                        },
                        "CacheClusterId": "redis12-002"
                    }
                ],
                "NodeGroupId": "0001",
                "PrimaryEndpoint": {
                    "Port": 6379,
                    "Address": "redis12.v5r9dc.ng.0001.usw2.cache.amazonaws.com"
```

```
                }
            }
        ],
        "ReplicationGroupId": "redis12",
        "SnapshotRetentionLimit": 1,
        "AutomaticFailover": "enabling",
        "MultiAZ": "enabled",
        "SnapshotWindow": "07:00-08:00",
        "SnapshottingClusterId": "redis12-002",
        "MemberClusters": [
            "redis12-001",
            "redis12-002"
        ],
        "PendingModifiedValues": {}
    }
}
```

Para obtener más información, consulte los temas siguientes en la Referencia de los comandos de la AWS CLI:

- create-cache-cluster
- create-replication-group
- modify-replication-group en la Referencia de los comandos de la AWS CLI.

Habilitación de Multi-AZ (API de ElastiCache)

En el siguiente ejemplo de código se utiliza la API de ElastiCache a fin de habilitar Multi-AZ para el grupo de reproducción `redis12`.

> **Note**
>
> Para usar este ejemplo, el grupo de reproducción `redis12` debe existir y tener al menos una réplica de lectura disponible.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=ModifyReplicationGroup
    &ApplyImmediately=true
    &AutoFailover=true
    &MultiAZEnabled=true
```

```
    &ReplicationGroupId=redis12
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20140401T192317Z
    &X-Amz-Credential=<credential>
```

Para obtener más información, consulte los siguientes temas en la Referencia de la API de ElastiCache:

- CreateCacheCluster
- CreateReplicationGroup
- ModifyReplicationGroup

Escenarios de error con respuestas de Multi-AZ

Antes de la presentación de Multi-AZ, ElastiCache detectaba y sustituía los nodos con error de un clúster al recrear y reaprovisionar el nodo que producía error. Si habilita Multi-AZ, un nodo principal que produce error conmuta por error a la réplica con el menor retraso de reproducción. La réplica seleccionada se promocionará automáticamente a la principal, lo cual es mucho más rápido que crear y reaprovisionar un nuevo nodo principal. Este proceso suele tardar tan solo unos segundos hasta que se puede escribir de nuevo en el clúster.

Cuando Multi-AZ se encuentra habilitado, ElastiCache monitorea continuamente el estado del nodo primario. Si se produce un error en el nodo principal, se realiza una de las siguientes acciones en función del tipo de error.

Temas

- Escenarios de error cuando solo se produce un error en el nodo principal
- Escenarios de error cuando el nodo primario y algunas réplicas de lectura producen un error
- Escenarios de error cuando se produce un error en todo el clúster

Escenarios de error cuando solo se produce un error en el nodo principal

Si solo se produce un error en el nodo principal, la réplica de lectura con el menor retardo de reproducción se promociona al clúster principal. A continuación, se crea una réplica de lectura de reemplazo y se aprovisiona en la misma zona de disponibilidad que el principal ha producido un error.

Cuando solo se produce un error en el nodo primario, Multi-AZ de ElastiCache realiza lo siguiente:

1. El nodo principal con error se desconecta (sin conexión).
2. La réplica de lectura con el mínimo retardo de reproducción se promociona a nodo principal.

   Las operaciones de escritura se pueden reanudar tan pronto como se haya completado el proceso de promoción, por lo general, en tan solo unos segundos. Si la aplicación escribe en el punto de conexión principal, no es necesario cambiar el punto de enlace para escrituras o lecturas. ElastiCache propaga el nombre de DNS de la réplica promocionada.
3. Una réplica de lectura de reemplazo se lanza y aprovisiona.

   La réplica de lectura de reemplazo se lanza en la zona de disponibilidad en la que estaba el nodo principal con error, por lo que se mantiene la distribución de los nodos.

4. Las réplicas se sincronizan con el nuevo nodo principal.

Una vez que la nueva réplica esté disponible, tenga en cuenta estos efectos:

- Punto de enlace principal: no debe realizar cambios en su aplicación, ya que el nombre de DNS del nodo primario nuevo se propagará al punto de conexión principal.
- Punto de enlace de lectura: el punto de conexión del lector se actualiza de forma automática para apuntar a los nodos de réplica nuevos.

Para obtener información acerca de la búsqueda de los puntos de conexión de un clúster, consulte los temas siguientes:

- Búsqueda de puntos de conexión de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) (consola)
- Búsqueda de los puntos de conexión de los grupos de replicación de Valkey o Redis OSS (AWS CLI)
- Búsqueda de puntos de conexión de grupos de replicación de Valkey o Redis OSS (API de ElastiCache)

Escenarios de error cuando el nodo primario y algunas réplicas de lectura producen un error

Si se produce un error en el nodo principal y en al menos una réplica, la réplica disponible con el menor retardo de reproducción se promocionará al clúster principal. Las nuevas réplicas de lectura también se crean y se aprovisionan en las mismas zonas de disponibilidad que las de los nodos con error y que la réplica que se promocionó a nodo principal.

Cuando el nodo primario y algunas réplicas de lectura producen un error, Multi-AZ de ElastiCache realiza lo siguiente:

1. El nodo principal y las réplicas de lectura con error se desconectan.
2. La réplica disponible con el mínimo retardo de reproducción se promociona a nodo principal.

   Las operaciones de escritura se pueden reanudar tan pronto como se haya completado el proceso de promoción, por lo general, en tan solo unos segundos. Si la aplicación escribe en el punto

de conexión principal, no es necesario cambiar el punto de enlace para escrituras. ElastiCache propaga el nombre de DNS de la réplica promocionada.

3. Las réplicas de reemplazo se crean y se aprovisionan.

   Las réplicas de reemplazo se crean en las zonas de disponibilidad de los nodos con error para, de este modo, conservar la distribución de los nodos.

4. Todos los clústeres se sincronizan con el nodo principal.

Debe realizar los siguientes cambios en su aplicación una vez que los nuevos nodos estén disponibles:

- Punto de conexión principal: no realice cambios en su aplicación. El nombre de DNS del nuevo nodo principal se propaga al punto de conexión principal.
- Punto de conexión de lectura: el punto de enlace de lectura se actualiza de forma automática para apuntar a los nodos de réplica nuevos.

Para obtener información acerca de la búsqueda de los puntos de enlace de un grupo de replicación, consulte los temas siguientes:

- Búsqueda de puntos de conexión de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) (consola)
- Búsqueda de los puntos de conexión de los grupos de replicación de Valkey o Redis OSS (AWS CLI)
- Búsqueda de puntos de conexión de grupos de replicación de Valkey o Redis OSS (API de ElastiCache)

Escenarios de error cuando se produce un error en todo el clúster

Si el error es general, todos los nodos se volverán a crear y a aprovisionar en las mismas zonas de disponibilidad que las de los nodos originales.

En esta situación, se perderán todos los datos del clúster debido al error de todos los nodos del clúster. Este tipo de error no suele producirse con frecuencia.

Cuando se produce un error en todo el clúster, Multi-AZ de ElastiCache realiza lo siguiente:

1. El nodo principal y las réplicas de lectura se desconectan.
2. Se crea y se aprovisiona un nodo principal de reemplazo.
3. Las réplicas de reemplazo se crean y se aprovisionan.

   Los reemplazos se crean en las zonas de disponibilidad de los nodos con error para, de este modo, conservar la distribución de los nodos.

   Puesto que el error ha afectado a la totalidad del clúster, los datos se perderán y los nuevos nodos se crean vacíos.

Puesto que cada uno de los nodos de reemplazo tendrán el mismo punto de conexión que el nodo al que reemplacen, no es necesario realizar ningún cambio de punto de conexión en su aplicación.

Para obtener información acerca de la búsqueda de los puntos de enlace de un grupo de replicación, consulte los temas siguientes:

- Búsqueda de puntos de conexión de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) (consola)
- Búsqueda de los puntos de conexión de los grupos de replicación de Valkey o Redis OSS (AWS CLI)
- Búsqueda de puntos de conexión de grupos de replicación de Valkey o Redis OSS (API de ElastiCache)

Recomendamos que cree el nodo principal y las réplicas de lectura en distintas zonas de disponibilidad para incrementar el nivel de tolerancia a errores.

Prueba de la conmutación por error automática

Después de habilitar la conmutación por error automática, puede probarla mediante la consola de ElastiCache, la AWS CLI y la API de ElastiCache.

Cuando realice las pruebas, tenga en cuenta lo siguiente:

- Esta operación se puede utilizar para probar la conmutación por error automática en un máximo de 15 particiones (denominadas grupos de nodos en la AWS CLI y la API de ElastiCache) en un periodo continuo de 24 horas.
- Si realiza una llamada a esta operación en fragmentos de distintos clústeres (denominados grupos de reproducción en la API y la CLI), podrá realizar las llamadas de forma simultánea.
- En algunos casos, es posible que llame a esta operación varias veces en particiones diferentes del mismo grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado). En tales casos, la sustitución del primer nodo debe completarse antes de que se pueda realizar una llamada posterior.
- Para determinar si la sustitución del nodo se ha completado, verifique los eventos mediante la consola de Amazon ElastiCache, la AWS CLI o la API de ElastiCache. Busque los eventos relacionados con la conmutación por error automática que se indican a continuación por orden de incidencia:
  1. Mensaje del grupo de replicación: `Test Failover API called for node group <node-group-id>`
  2. Mensaje del clúster de caché: `Failover from primary node <primary-node-id> to replica node <node-id> completed`
  3. Mensaje del grupo de replicación: `Failover from primary node <primary-node-id> to replica node <node-id> completed`
  4. Mensaje del clúster de caché: `Recovering cache nodes <node-id>`
  5. Mensaje del clúster de caché: `Finished recovery for cache nodes <node-id>`

  Para obtener más información, consulte los siguientes temas:
  - Visualización de eventos de ElastiCache en la Guía del usuario de ElastiCache
  - DescribeEvents en la Referencia de la API de ElastiCache
  - describe-events en la Referencia de los comandos de la AWS CLI.
- Esta API está diseñada para probar el comportamiento de la aplicación en caso de conmutación por error de ElastiCache. No está diseñado para ser una herramienta operativa para iniciar una conmutación por error para solucionar un problema con el clúster. Además, en determinadas

condiciones, como los acontecimientos operacionales a gran escala, AWS puede bloquear esta API.

Temas

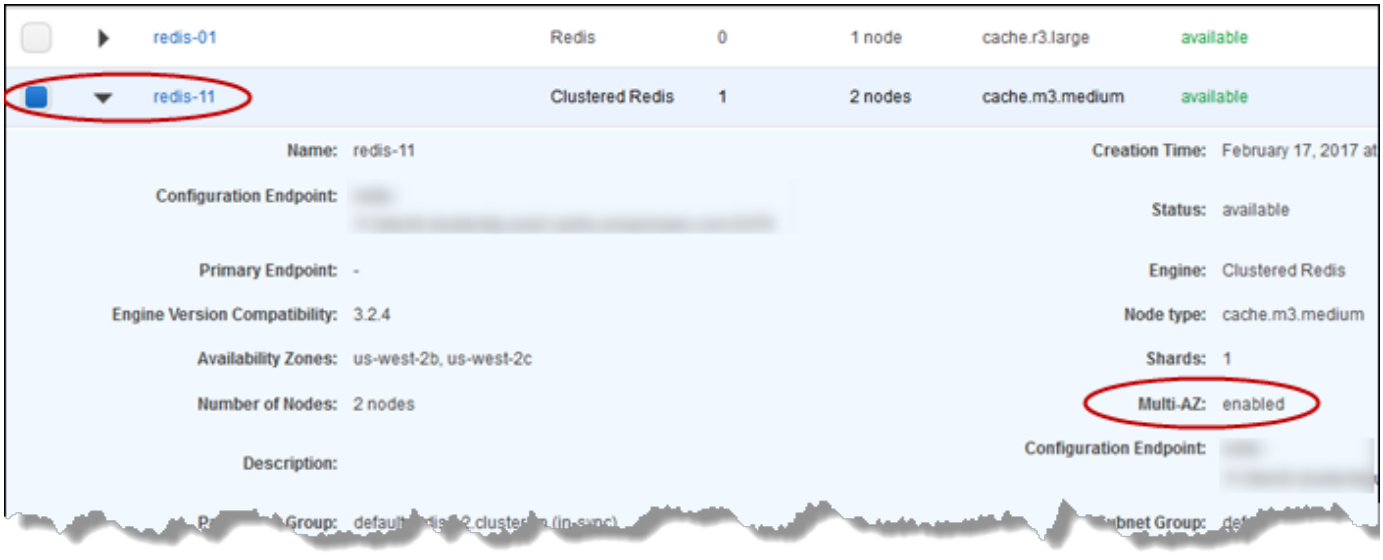
- Prueba de la conmutación por error automática mediante la Consola de administración de AWS
- Prueba de la conmutación por error automática mediante la AWS CLI
- Prueba de la conmutación por error automática mediante la API de ElastiCache

Prueba de la conmutación por error automática mediante la Consola de administración de AWS

Utilice el procedimiento siguiente para probar la conmutación por error automática con la consola.

Para probar la conmutación por error automática

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Valkey o Redis OSS.
3. En la lista de clústeres, elija el cuadro situado a la izquierda del clúster que desea probar. El clúster debe tener al menos un nodo de réplica de lectura.
4. En el área Details, asegúrese de que este clúster tiene habilitadas Multi-AZ. Si el clúster no tiene habilitado Multi-AZ, elija un clúster distinto o modifique este clúster para habilitar Multi-AZ. Para obtener más información, consulte Uso de la Consola de administración de AWS de ElastiCache.



5. Para Valkey o Redis OSS (modo de clúster deshabilitado), elija el nombre del clúster.

Para Valkey o Redis OSS (modo de clúster habilitado), realice lo siguiente:

a. Elija el nombre del clúster.
b. En la página Shards, elija el nombre del fragmento (denominado grupo de nodos en la API y la CLI) en el que desea probar la conmutación por error.

6. En la página Nodos, elija Failover Primary.
7. Elija Continue para realizar la conmutación por error al nodo principal, o bien Cancel para cancelar la operación y no realizar la conmutación por error al nodo principal.

   Durante el proceso de conmutación por error, la consola seguirá mostrando el estado del nodo como disponible. Para realizar un seguimiento del progreso de la prueba de la conmutación por error, elija Events en el panel de navegación de la consola. En la pestaña Eventos, consulte los eventos que indican que la conmutación por error se ha iniciado (`Test Failover API called`) y completado (`Recovery completed`).

Prueba de la conmutación por error automática mediante la AWS CLI

Puede probar la conmutación por error automática en cualquier clúster habilitado de Multi-AZ mediante la operación de la AWS CLI `test-failover`.

Parámetros

- `--replication-group-id`: obligatorio. Grupo de reproducción (en la consola, clúster) que se va a comprobar.
- `--node-group-id`: obligatorio. Nombre del grupo de nodos en el que desea probar la conmutación por error automática. Puede probar un máximo de 15 grupos de nodos en un periodo continuo de 24 horas.

El siguiente ejemplo utiliza la AWS CLI para probar la conmutación por error automática en el grupo de nodos `redis00-0003` del clúster de Valkey o Redis OSS (modo de clúster habilitado) `redis00`.

Example Pruebe la conmutación por error automática

Para Linux, macOS o Unix:

```
aws elasticache test-failover \
```

```
   --replication-group-id redis00 \
   --node-group-id redis00-0003
```

Para Windows:

```
aws elasticache test-failover ^
   --replication-group-id redis00 ^
   --node-group-id redis00-0003
```

La salida del comando anterior es similar a la siguiente.

```
{
    "ReplicationGroup": {
        "Status": "available",
        "Description": "1 shard, 3 nodes (1 + 2 replicas)",
        "NodeGroups": [
            {
                "Status": "available",
                "NodeGroupMembers": [
                    {
                        "CurrentRole": "primary",
                        "PreferredAvailabilityZone": "us-west-2c",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Port": 6379,
                            "Address":
 "redis1x3-001.7ekv3t.0001.usw2.cache.amazonaws.com"
                        },
                        "CacheClusterId": "redis1x3-001"
                    },
                    {
                        "CurrentRole": "replica",
                        "PreferredAvailabilityZone": "us-west-2a",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Port": 6379,
                            "Address":
 "redis1x3-002.7ekv3t.0001.usw2.cache.amazonaws.com"
                        },
                        "CacheClusterId": "redis1x3-002"
                    },
                    {
                        "CurrentRole": "replica",
```

```
                        "PreferredAvailabilityZone": "us-west-2b",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Port": 6379,
                            "Address":
 "redis1x3-003.7ekv3t.0001.usw2.cache.amazonaws.com"
                        },
                        "CacheClusterId": "redis1x3-003"
                    }
                ],
                "NodeGroupId": "0001",
                "PrimaryEndpoint": {
                    "Port": 6379,
                    "Address": "redis1x3.7ekv3t.ng.0001.usw2.cache.amazonaws.com"
                }
            }
        ],
        "ClusterEnabled": false,
        "ReplicationGroupId": "redis1x3",
        "SnapshotRetentionLimit": 1,
        "AutomaticFailover": "enabled",
        "MultiAZ": "enabled",
        "SnapshotWindow": "11:30-12:30",
        "SnapshottingClusterId": "redis1x3-002",
        "MemberClusters": [
            "redis1x3-001",
            "redis1x3-002",
            "redis1x3-003"
        ],
        "CacheNodeType": "cache.m3.medium",
        "DataTiering": "disabled",
        "PendingModifiedValues": {}
    }
}
```

Para realizar un seguimiento del progreso de la conmutación por error, use la operación de la AWS CLI `describe-events`.

Para obtener más información, consulte los siguientes temas:

- test-failover en la Referencia de los comandos de la AWS CLI.
- describe-events en la Referencia de los comandos de la AWS CLI.

Prueba de la conmutación por error automática mediante la API de ElastiCache

Puede probar la conmutación por error automática en cualquier clúster habilitado con Multi-AZ mediante la operación de la API de ElastiCache `TestFailover`.

Parámetros

- `ReplicationGroupId`: obligatorio. El grupo de reproducción (en la consola, clúster) que se va a comprobar.
- `NodeGroupId`: obligatorio. Nombre del grupo de nodos en el que desea probar la conmutación por error automática. Puede probar un máximo de 15 grupos de nodos en un periodo continuo de 24 horas.

El ejemplo siguiente comprueba la conmutación por error automática en el grupo de nodos `redis00-0003` del grupo de reproducción (clúster, en la consola) `redis00`.

Example Prueba de la conmutación por error automática

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=TestFailover
    &NodeGroupId=redis00-0003
    &ReplicationGroupId=redis00
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20140401T192317Z
    &X-Amz-Credential=<credential>
```

Para realizar un seguimiento del progreso de la conmutación por error, utilice la operación de la API de ElastiCache `DescribeEvents`.

Para obtener más información, consulte los siguientes temas:

- TestFailover en la Referencia de la API de ElastiCache
- DescribeEvents en la Referencia de la API de ElastiCache

Limitaciones en multi-AZ

Tenga en cuenta las siguientes limitaciones para multi-AZ:

- Multi-AZ se admite en Valkey y en Redis OSS versión 2.8.6 y posteriores.
- Multi-AZ no se admite en los tipos de nodos T1.
- La replicación de Valkey y Redis OSS es asíncrona. Por lo tanto, cuando un nodo principal realiza una conmutación por error a una réplica, se puede perder una pequeña cantidad de datos debido al retraso de reproducción.

  Cuando elige que la réplica se promueva a principal, ElastiCache selecciona la réplica con el menor retardo de replicación. En otras palabras, elija la réplica más actual. Esto ayuda a minimizar la cantidad de datos perdidos. La réplica que tiene el menor retardo de reproducción puede estar en la misma zona de disponibilidad que el nodo principal con error o en otra.
- Si promueve manualmente réplicas de lectura a principal en clústeres de Valkey o Redis OSS con el modo de clúster deshabilitado, solo podrá hacerlo cuando multi-AZ y la conmutación por error automática estén deshabilitadas. Para promocionar una réplica de lectura a principal, siga estos pasos:
  1. Deshabilite Multi-AZ en el clúster.
  2. Deshabilite la conmutación por error automática en el clúster. Puede hacerlo a través de la consola. Para ello, desactive la casilla de verificación Conmutación por error automática del grupo de replicación. También puede hacerlo utilizando la AWS CLI y estableciendo la propiedad `AutomaticFailoverEnabled` en `false` al llamar a la operación `ModifyReplicationGroup`.
  3. Promocione la réplica de lectura a principal.
  4. Vuelva a habilitar Multi-AZ.
- Multi-AZ de ElastiCache para Redis OSS y archivos de solo adición (AOF) son mutuamente excluyentes. Si habilita una opción, no puede habilitar la otra.
- El error de un nodo puede ser provocado por el improbable caso de que deje de funcionar una zona de disponibilidad completa. En este caso, la réplica que sustituye a la principal con error se creará únicamente si hay una copia de seguridad de la zona de disponibilidad. Por ejemplo, considere la posibilidad de un grupo de reproducción con el principal en AZ-a y las réplicas en AZ-b y AZ-c. Si el principal falla, la réplica con el menor retardo de reproducción se promociona al clúster principal. A continuación, ElastiCache crea una réplica nueva en la AZ-a (en la que se encontró la réplica principal con el error) solo si se ha realizado una copia de seguridad de AZ-a y se encuentra disponible.

- Un reinicio iniciado por un cliente de un principal no desencadena la conmutación por error automática. Otros reinicios y errores sí activan la conmutación por error automática.
- Cuando se reinicia el principal, sus datos se borran cuando vuelve a estar en línea. Cuando las réplicas de lectura ven el clúster principal borrado, borran sus copias de los datos, lo que provoca una pérdida de datos.
- Después de la promoción de una réplica de lectura, las otras réplicas se sincronizan con el nuevo principal. Después de la sincronización inicial, el contenido de las réplicas se elimina y sincronizan los datos del nuevo principal. Este proceso de sincronización provoca una breve interrupción, durante la cual no se puede acceder a las réplicas. El proceso de sincronización también provoca un aumento de carga temporal en el principal mientras se sincroniza con las réplicas. Este comportamiento es nativo de Valkey y Redis OSS y no es exclusivo para multi-AZ de ElastiCache. Para obtener más información sobre este comportamiento, consulte Replication en el sitio web de Valkey.

Important

En Valkey 7.2.6 y versiones posteriores o Redis OSS versión 2.8.22 y versiones posteriores, no se pueden crear réplicas externas.
Para las versiones de Redis OSS anteriores a la 2.8.22, se recomienda que no conecte una réplica externa a un clúster de ElastiCache habilitado para multi-AZ. Se trata de una configuración no admitida que puede crear problemas que impiden a ElastiCache realizar la conmutación por error y la recuperación correctamente. Para conectar una réplica externa a un clúster de ElastiCache, asegúrese de que multi-AZ no se encuentre habilitado antes de realizar la conexión.

Cómo se implementan la sincronización y la copia de seguridad

Todas las versiones compatibles de Valkey o Redis OSS admiten la copia de seguridad y la sincronización entre el nodo principal y el nodo de réplica. Sin embargo, la forma en la que se implementan la copia de seguridad y la sincronización varía en función de la versión.

Versión 2.8.22 de Redis OSS y posteriores

En las versiones 2.8.22 y posteriores, la replicación de Redis OSS se realiza con uno de dos métodos. Para obtener más información, consulte Versiones de Redis OSS anteriores a la 2.8.22 y Instantánea y restauración.

Durante el proceso sin ramificaciones, si la carga de escritura es elevada, las operaciones de escritura en el clúster se retrasan para garantizar que no se acumulan demasiados cambios que impidan el éxito del proceso de instantánea.

Versiones de Redis OSS anteriores a la 2.8.22

La copia de seguridad y la sincronización de Redis OSS en versiones anteriores a la 2.8.22 es un proceso que consta de tres pasos.

1. Las ramificaciones y en el proceso en segundo plano, serializan los datos del clúster en disco. Esto crea una instantánea puntual.
2. En primer plano, se acumula un log de cambios en el búfer de salida del cliente.

   > **Important**
   >
   > Si el log de cambios supera el tamaño del búfer de salida del cliente, la copia de seguridad o la sincronización fallarán. Para obtener más información, consulte Forma de garantizar que dispone de memoria suficiente para crear una instantánea de Valkey o Redis OSS.

3. Por último, transmita los datos de caché y, a continuación, el log de cambios al nodo de réplica.

Creación de un grupo de replicación de Valkey o Redis OSS

Dispone de las siguientes opciones para crear un clúster con nodos de réplica. Uno se aplica cuando ya tiene un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) disponible no asociado con un clúster que tiene réplicas que se van a utilizar como nodo principal. El otro se aplica cuando necesita crear un nodo principal con el clúster y réplicas de lectura. Actualmente, se debe crear un clúster de Valkey o Redis OSS (modo de clúster habilitado) desde cero.

Opción 1: Creación de un grupo de reproducción mediante un clúster existente

Utilice esta opción para utilizar un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) de un solo nodo. Podrá especificar este nodo existente como el nodo primario en el clúster nuevo y, a continuación, agregar individualmente de 1 a 5 réplicas de lectura al clúster. Si el clúster existente está activo, las réplicas de lectura se sincronizan con él conforme se creen. Consulte Creación de un grupo de reproducción mediante un clúster existente.

> **Important**
>
> No puede crear un clúster de Valkey o Redis OSS (modo de clúster habilitado) con un clúster existente. Para crear un clúster de Valkey o Redis OSS (modo de clúster habilitado) (API/CLI: grupo de replicación) mediante la consola de ElastiCache, consulte Creación de un clúster de Valkey o Redis OSS (modo de clúster habilitado) (consola).

Opción 2: Creación de un grupo de replicación de Valkey o Redis OSS desde cero

Utilice esta opción si aún no dispone de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) para utilizar como nodo principal, o si desea crear un clúster de Valkey o Redis OSS (modo de clúster habilitado). Consulte Creación de un grupo de replicación de Valkey o Redis OSS desde cero.

Creación de un grupo de reproducción mediante un clúster existente

El siguiente procedimiento añade un grupo de replicación a su clúster de un solo nodo de Valkey o Redis OSS (modo de clúster deshabilitado), que es necesario para actualizar el clúster a la última versión de Valkey. Se trata de un procedimiento in situ que no implica ningún tiempo de inactividad ni pérdida de datos. Al crear un grupo de replicación para un clúster de un solo nodo, el nodo del clúster se convierte en el nodo principal del nuevo clúster. Si no tiene un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) que pueda utilizar como el nuevo principal del clúster; consulte Creación de un grupo de replicación de Valkey o Redis OSS desde cero.

Un clúster disponible es un clúster de Valkey o Redis OSS de un solo nodo existente. Actualmente, Valkey o Redis OSS (modo de clúster habilitado) no admiten la creación de un clúster con réplicas mediante un clúster de un solo nodo disponible. Si desea crear un clúster de Valkey o Redis OSS (modo de clúster habilitado), consulte Creación de un clúster de Valkey o Redis OSS (modo de clúster habilitado) (consola).

Creación de un grupo de replicación mediante un clúster existente (consola)

Consulte el tema Uso de la Consola de administración de AWS de ElastiCache.

Creación de un grupo de replicación mediante un clúster de Valkey o Redis OSS disponible (AWS CLI)

Hay dos pasos para crear un grupo de replicación con réplicas de lectura al utilizar un clúster de caché de Valkey o Redis OSS disponible para el nodo principal cuando se utiliza la AWS CLI.

Al utilizar la AWS CLI, se crea un grupo de reproducción especificando el nodo independiente disponible como el nodo principal del clúster, `--primary-cluster-id` y el número de nodos que desea en el clúster con el comando `create-replication-group` de la CLI. Incluya los siguientes parámetros.

--replication-group-id

El nombre del grupo de replicación que va a crear. El valor de este parámetro se usa como la base de los nombres de los nodos añadidos con un número de tres dígitos secuencial añadido al final de `--replication-group-id`. Por ejemplo, `sample-repl-group-001`.

Las restricciones de nomenclatura de los grupos de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) son las siguientes:

- Deben contener entre 1 y 40 caracteres alfanuméricos o guiones.

- Deben comenzar por una letra.
- No pueden contener dos guiones consecutivos.
- No pueden terminar con un guion.

--replication-group-description

Descripción del grupo de replicación.

--num-node-groups

El número de nodos que desea en este clúster. Este valor incluye el nodo principal. Este parámetro tiene un valor máximo de seis.

--primary-cluster-id

El nombre del nodo del clúster de Valkey o Redis OSS (modo de clúster deshabilitado) disponible que desea que sea el nodo principal en este grupo de replicación.

El siguiente comando crea el grupo de replicación `sample-repl-group` mediante el clúster de Valkey o Redis OSS (modo de clúster deshabilitado) disponible `redis01` como el nodo principal del grupo de replicación. Crea dos nuevos nodos que son réplicas de lectura. La configuración de `redis01` (es decir, grupo de parámetros, grupo de seguridad, tipo de nodo, versión del motor, etc.) se aplicará a todos los nodos del grupo de reproducción.

Para Linux, macOS o Unix:

```
aws elasticache create-replication-group \
   --replication-group-id sample-repl-group \
   --replication-group-description "demo cluster with replicas" \
   --num-cache-clusters 3 \
   --primary-cluster-id redis01
```

Para Windows:

```
aws elasticache create-replication-group ^
   --replication-group-id sample-repl-group ^
   --replication-group-description "demo cluster with replicas" ^
   --num-cache-clusters 3 ^
   --primary-cluster-id redis01
```

Para obtener información adicional y parámetros que puede querer utilizar, consulte el tema AWS CLI de la create-replication-group.

A continuación, añada réplicas de lectura al grupo de replicación.

Después de haber creado el grupo de reproducción, agregue de una a cinco réplicas de lectura con el comando `create-cache-cluster`, asegurándose de incluir los siguientes parámetros.

--cache-cluster-id

El nombre del clúster que va a añadir al grupo de replicación.

Las restricciones para la asignación de nombres de clúster son las siguientes:

- Deben contener entre 1 y 40 caracteres alfanuméricos o guiones.
- Deben comenzar por una letra.
- No pueden contener dos guiones consecutivos.
- No pueden terminar con un guion.

--replication-group-id

Es el nombre del grupo de replicación al que va a añadir este clúster.

Repita este comando para cada réplica de lectura que desee agregar al grupo de reproducción, cambiando únicamente el valor del parámetro `--cache-cluster-id`.

> **Note**
>
> Recuerde que un grupo de reproducción no puede tener más de cinco réplicas de lectura. Si intenta agregar una réplica de lectura a un grupo de reproducción que ya tenga cinco réplicas de lectura hará que la operación falle.

El siguiente código añade la réplica de lectura `my-replica01` al grupo de replicación `sample-repl-group`. La configuración del clúster principal (grupo de parámetros, grupo de seguridad, tipo de nodo, etc.) se aplicará a los nodos que se agreguen al grupo de reproducción.

Para Linux, macOS o Unix:

```
aws elasticache create-cache-cluster \
   --cache-cluster-id my-replica01 \
   --replication-group-id sample-repl-group
```

Para Windows:

```
aws elasticache create-cache-cluster ^
   --cache-cluster-id my-replica01 ^
   --replication-group-id sample-repl-group
```

El resultado de este comando será algo similar a lo que se muestra a continuación.

```
{
    "ReplicationGroup": {
        "Status": "creating",
        "Description": "demo cluster with replicas",
        "ClusterEnabled": false,
        "ReplicationGroupId": "sample-repl-group",
        "SnapshotRetentionLimit": 1,
        "AutomaticFailover": "disabled",
        "SnapshotWindow": "00:00-01:00",
        "SnapshottingClusterId": "redis01",
        "MemberClusters": [
            "sample-repl-group-001",
            "sample-repl-group-002",
            "redis01"
        ],
        "CacheNodeType": "cache.m4.large",
        "DataTiering": "disabled",
        "PendingModifiedValues": {}
    }
}
```

Para obtener más información, consulte los temas de AWS CLI:

- create-replication-group
- modify-replication-group

Adición de réplicas a un clúster independiente de Valkey o Redis OSS (modo de clúster deshabilitado) (API de ElastiCache)

Al utilizar la API de ElastiCache, se crea un grupo de reproducción al especificar el nodo independiente disponible como el nodo primario del clúster, `PrimaryClusterId` y el número de nodos que se desean para el clúster con el comando de la CLI, `CreateReplicationGroup`. Incluya los siguientes parámetros.

ReplicationGroupId

El nombre del grupo de replicación que va a crear. El valor de este parámetro se usa como la base de los nombres de los nodos añadidos con un número de tres dígitos secuencial añadido al final de `ReplicationGroupId`. Por ejemplo, `sample-repl-group-001`.

Las restricciones de nomenclatura de los grupos de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) son las siguientes:

- Deben contener entre 1 y 40 caracteres alfanuméricos o guiones.
- Deben comenzar por una letra.
- No pueden contener dos guiones consecutivos.
- No pueden terminar con un guion.

ReplicationGroupDescription

Descripción del clúster con réplicas.

NumCacheClusters

El número de nodos que desea en este clúster. Este valor incluye el nodo principal. Este parámetro tiene un valor máximo de seis.

PrimaryClusterId

El nombre del clúster de Valkey o Redis OSS (modo de clúster deshabilitado) disponible que desea que sea el nodo principal en este clúster.

El siguiente comando crea el clúster con réplicas `sample-repl-group` mediante el clúster de Valkey o Redis OSS (modo de clúster deshabilitado) `redis01` disponible como el nodo principal del grupo de replicación. Crea dos nuevos nodos que son réplicas de lectura. La configuración de `redis01` (es decir, grupo de parámetros, grupo de seguridad, tipo de nodo, versión del motor, etc.) se aplicará a todos los nodos del grupo de reproducción.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=CreateReplicationGroup
   &Engine=redis
   &EngineVersion=6.0
   &ReplicationGroupDescription=Demo%20cluster%20with%20replicas
   &ReplicationGroupId=sample-repl-group
   &PrimaryClusterId=redis01
   &Version=2015-02-02
```

```
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &X-Amz-Credential=<credential>
```

Para obtener más información, consulte los temas de APL de ElastiCache:

- CreateReplicationGroup
- ModifyReplicationGroup

A continuación, añada réplicas de lectura al grupo de replicación.

Después de haber creado el grupo de reproducción, agregue de una a cinco réplicas de lectura con la operación `CreateCacheCluster`, asegurándose de incluir los siguientes parámetros.

CacheClusterId

El nombre del clúster que va a añadir al grupo de replicación.

Las restricciones para la asignación de nombres de clúster son las siguientes:

- Deben contener entre 1 y 40 caracteres alfanuméricos o guiones.
- Deben comenzar por una letra.
- No pueden contener dos guiones consecutivos.
- No pueden terminar con un guion.

ReplicationGroupId

Es el nombre del grupo de replicación al que va a añadir este clúster.

Repita esta operación para cada réplica de lectura que desee agregar al grupo de reproducción, cambiando únicamente el valor del parámetro `CacheClusterId`.

El siguiente código agrega la réplica de lectura `myReplica01` al grupo de reproducción `myReplGroup`. La configuración del clúster principal (grupo de parámetros, grupo de seguridad, tipo de nodo, etc.) se aplicará a los nodos que se agreguen al grupo de reproducción.

```
https://elasticache.us-west-2.amazonaws.com/
 ?Action=CreateCacheCluster
 &CacheClusterId=myReplica01
```

```
&ReplicationGroupId=myReplGroup
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&Version=2015-02-02
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Credential=[your-access-key-id]/20150202/us-west-2/elasticache/aws4_request
&X-Amz-Date=20150202T170651Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=[signature-value]
```

Para obtener información adicional y parámetros que puede querer utilizar, consulte el tema de la API de ElastiCache CreateCacheCluster.

Creación de un grupo de replicación de Valkey o Redis OSS desde cero

A continuación, explicaremos cómo crear un grupo de replicación de Valkey o Redis OSS sin utilizar un clúster de Valkey o Redis OSS existente como principal. Puede crear un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) o Valkey o Redis OSS (modo de clúster habilitado) desde cero mediante la consola de ElastiCache, la AWS CLI o la API de ElastiCache.

Antes de continuar, decida si desea crear un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) o Valkey o Redis OSS (modo de clúster habilitado). Si desea obtener orientación para tomar su decisión, consulte Replicación: Valkey y Redis OSS en modo de clúster deshabilitado o habilitado.

Temas

- Creación de un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) desde cero
- Creación de un grupo de replicación en Valkey o Redis OSS (modo de clúster habilitado) desde cero

Creación de un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) desde cero

Puede crear un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) desde cero mediante la consola de ElastiCache, la AWS CLI o la API de ElastiCache. Un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) siempre tiene un grupo de nodos, un clúster principal y hasta cinco réplicas de lectura. Los grupos de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) no admiten la partición de los datos.

> **Note**
>
> El límite de nodos o de particiones puede aumentarse hasta un máximo de 500 por clúster. Para solicitar un aumento del límite, consulte AWS Service Limits e incluya el tipo de instancias en la solicitud.

Para crear un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) desde cero, siga uno de estos métodos:

Creación de un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) desde cero (AWS CLI)

El siguiente procedimiento crea un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) mediante la AWS CLI.

Al crear un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) desde cero, debe crear el grupo de replicación y todos sus nodos con una sola llamada al comando `create-replication-group` de la AWS CLI. Incluya los siguientes parámetros.

--replication-group-id

El nombre del grupo de replicación que va a crear.

Las restricciones de nomenclatura de los grupos de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) son las siguientes:

- Deben contener entre 1 y 40 caracteres alfanuméricos o guiones.
- Deben comenzar por una letra.
- No pueden contener dos guiones consecutivos.
- No pueden terminar con un guion.

--replication-group-description

Descripción del grupo de replicación.

--num-cache-clusters

El número de nodos que desea crear con este grupo de reproducción, combinando el principal y las réplicas de lectura.

Si habilita las Multi-AZ (`--automatic-failover-enabled`), el valor de `--num-cache-clusters` debe ser de al menos 2.

--cache-node-type

El tipo de nodo para cada nodo en el grupo de replicación.

ElastiCache admite los siguientes tipos de nodos. En general, los tipos de la generación actual ofrecen más capacidad de memoria y potencia de computación a menor costo comparados con sus equivalentes de generaciones previas.

Para obtener más información sobre los detalles del rendimiento de cada tipo de nodo, consulte Tipos de instancia de Amazon EC2.

--habilitado para organización de datos en niveles

Establezca este parámetro si está utilizando un tipo de nodo r6gd. Si no quiere la organización de datos en niveles, defina `--no-data-tiering-enabled`. Para obtener más información, consulte Organización de datos en niveles en ElastiCache.

--cache-parameter-group

Especifique un grupo de parámetros que corresponda a su versión de motor. Si va a ejecutar Redis OSS 3.2.4 o una versión posterior, especifique el grupo de parámetros `default.redis3.2` o un grupo de parámetros derivado de `default.redis3.2` para crear un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado). Para obtener más información, consulte Parámetros de Valkey y Redis OSS.

--network-type

`ipv4`, `ipv6` o `dual-stack`. Si elige la doble pila, debe establecer el parámetro `--IpDiscovery` en `ipv4` o `ipv6`.

--engine

redis

--engine-version

Para conseguir el conjunto de características más completo, elija la última versión de motor.

Los nombres de los nodos se derivarán del nombre del grupo de reproducción anexando `-00#` al nombre del grupo de reproducción. Por ejemplo, al utilizar el nombre de grupo de reproducción `myReplGroup`, el nombre del principal será `myReplGroup-001`, y el de las réplicas de lectura, de `myReplGroup-002` a `myReplGroup-006`.

Si desea habilitar el cifrado en tránsito o en reposo en este grupo de replicación, agregue uno o estos dos parámetros, `--transit-encryption-enabled` y `--at-rest-encryption-enabled`, y ajústese a las siguientes condiciones.

- El grupo de replicación debe ejecutar la versión de Redis OSS 3.2.6 o 4.0.10.
- El grupo de reproducción debe crearse en una Amazon VPC.
- Debe incluirse también el parámetro `--cache-subnet-group`.
- Debe incluirse el parámetro `--auth-token` con el valor de cadena especificado por el cliente para el token AUTH (contraseña) que es necesario para realizar las operaciones en este grupo de replicación.

La siguiente operación crea un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) `sample-repl-group` con tres nodos, un nodo principal y dos réplicas.

Para Linux, macOS o Unix:

```
aws elasticache create-replication-group \
   --replication-group-id sample-repl-group \
   --replication-group-description "Demo cluster with replicas" \
   --num-cache-clusters 3 \
   --cache-node-type cache.m4.large \
   --engine redis
```

Para Windows:

```
aws elasticache create-replication-group ^
   --replication-group-id sample-repl-group ^
   --replication-group-description "Demo cluster with replicas" ^
   --num-cache-clusters 3 ^
```

```
    --cache-node-type cache.m4.large ^
    --engine redis
```

La salida de este comando es similar a la siguiente.

```
{
    "ReplicationGroup": {
        "Status": "creating",
        "Description": "Demo cluster with replicas",
        "ClusterEnabled": false,
        "ReplicationGroupId": "sample-repl-group",
        "SnapshotRetentionLimit": 0,
        "AutomaticFailover": "disabled",
        "SnapshotWindow": "01:30-02:30",
        "MemberClusters": [
            "sample-repl-group-001",
            "sample-repl-group-002",
            "sample-repl-group-003"
        ],
        "CacheNodeType": "cache.m4.large",
        "DataTiering": "disabled",
        "PendingModifiedValues": {}
    }
}
```

Para obtener información adicional y parámetros que puede querer utilizar, consulte el tema de la AWS CLI create-replication-group.

Creación de un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) desde cero (API de ElastiCache)

El siguiente procedimiento sirve para crear un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) mediante la API de ElastiCache.

Al crear un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) desde cero, debe crear el grupo de replicación y todos sus nodos con una sola llamada a la operación de la API de ElastiCache `CreateReplicationGroup`. Incluya los siguientes parámetros.

ReplicationGroupId

El nombre del grupo de replicación que va a crear.

Estas son las restricciones de nomenclatura de los grupos de replicación de Valkey o Redis OSS (modo de clúster habilitado):

- Deben contener entre 1 y 40 caracteres alfanuméricos o guiones.
- Deben comenzar por una letra.
- No pueden contener dos guiones consecutivos.
- No pueden terminar con un guion.

ReplicationGroupDescription

Su descripción del grupo de reproducción.

NumCacheClusters

El número total de nodos que desea crear con este grupo de reproducción, combinando el principal y las réplicas de lectura.

Si habilita las Multi-AZ (`AutomaticFailoverEnabled=true`), el valor de `NumCacheClusters` debe ser de al menos 2.

CacheNodeType

El tipo de nodo para cada nodo en el grupo de replicación.

ElastiCache admite los siguientes tipos de nodos. En general, los tipos de la generación actual ofrecen más capacidad de memoria y potencia de computación a menor costo comparados con sus equivalentes de generaciones previas.

Para obtener más información sobre los detalles del rendimiento de cada tipo de nodo, consulte Tipos de instancia de Amazon EC2.

--habilitado para organización de datos en niveles

Establezca este parámetro si está utilizando un tipo de nodo r6gd. Si no quiere la organización de datos en niveles, defina `--no-data-tiering-enabled`. Para obtener más información, consulte Organización de datos en niveles en ElastiCache.

CacheParameterGroup

Especifique un grupo de parámetros que corresponda a su versión de motor. Si va a ejecutar Redis OSS 3.2.4 o una versión posterior, especifique el grupo de parámetros `default.redis3.2` o un grupo de parámetros derivado de `default.redis3.2` para crear un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado). Para obtener más información, consulte Parámetros de Valkey y Redis OSS.

--network-type

`ipv4`, `ipv` o `dual-stack`. Si elige la doble pila, debe establecer el parámetro `--IpDiscovery` en `ipv4` o `ipv6`.

Motor

redis

EngineVersion

6.0

Los nombres de los nodos se derivarán del nombre del grupo de reproducción anexando `-00#` al nombre del grupo de reproducción. Por ejemplo, al utilizar el nombre de grupo de reproducción `myReplGroup`, el nombre del principal será `myReplGroup-001`, y el de las réplicas de lectura, de `myReplGroup-002` a `myReplGroup-006`.

Si desea habilitar el cifrado en tránsito o en reposo en este grupo de replicación, agregue uno o estos dos parámetros, `TransitEncryptionEnabled=true` y `AtRestEncryptionEnabled=true`, y ajústese a las siguientes condiciones.

- El grupo de replicación debe ejecutar la versión de Redis OSS 3.2.6 o 4.0.10.
- El grupo de reproducción debe crearse en una Amazon VPC.
- Debe incluirse también el parámetro `CacheSubnetGroup`.
- Debe incluirse el parámetro `AuthToken` con el valor de cadena especificado por el cliente para el token AUTH (contraseña) que es necesario para realizar las operaciones en este grupo de replicación.

La siguiente operación crea el grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) `myReplGroup` con tres nodos, un nodo principal y dos réplicas.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=CreateReplicationGroup
   &CacheNodeType=cache.m4.large
   &CacheParameterGroup=default.redis6.x
   &Engine=redis
   &EngineVersion=6.0
   &NumCacheClusters=3
   &ReplicationGroupDescription=test%20group
```

```
    &ReplicationGroupId=myReplGroup
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
    &X-Amz-Credential=<credential>
```

Para obtener información adicional y parámetros que puede querer utilizar, consulte el tema de la API de ElastiCache CreateReplicationGroup.

Creación de un grupo de replicación en Valkey o Redis OSS (modo de clúster habilitado) desde cero

Puede crear un clúster de Valkey o Redis OSS (modo de clúster habilitado) (API/CLI: grupo de replicación) con la consola de ElastiCache, la AWS CLI o la API de ElastiCache. Un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado) tiene de 1 a 500 particiones (API/CLI: grupos de nodos), un nodo principal en cada partición y hasta 5 réplicas de lectura en cada partición. Puede crear un clúster con un mayor número de particiones y un menor número de réplicas con un total de hasta 90 nodos por clúster. Esta configuración de clúster puede variar desde 90 particiones y 0 réplicas hasta 15 particiones y 5 réplicas, que es el número máximo de réplicas permitido.

El límite de nodos o particiones se puede aumentar a un máximo de 500 por clúster si la versión del motor de Valkey o Redis OSS es 5.0.6 o posteriores. Por ejemplo, puede elegir configurar un clúster de 500 nodos que oscila entre 83 particiones (uno primario y 5 réplicas por partición) y 500 particiones (único primario y sin réplicas). Asegúrese de que hay suficientes direcciones IP disponibles para acomodar el aumento. Algunos problemas comunes incluyen que las subredes del grupo de subredes tienen un rango CIDR demasiado pequeño o que otros clústeres comparten y utilizan considerablemente las subredes. Para obtener más información, consulte Creación de un grupo de subredes.

Para las versiones inferiores a 5.0.6, el límite es de 250 por clúster.

Para solicitar un aumento del límite, consulte AWS Service Limits y elija el tipo de límite Nodes per cluster per instance type (Nodos por clúster por tipo de instancias).

Creación de un clúster en Valkey o Redis OSS (modo de clúster habilitado)

- Creación de un clúster de Valkey o Redis OSS (modo de clúster habilitado) (consola)
- Creación de un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado) desde cero (AWS CLI)
- Creación de un grupo de replicación en Valkey o Redis OSS (modo de clúster habilitado) desde cero (API de ElastiCache)

Creación de un clúster de Valkey o Redis OSS (modo de clúster habilitado) (consola)

Para crear un clúster de Valkey o Redis OSS (modo de clúster habilitado), consulte Creación de un clúster de Valkey o Redis OSS (modo de clúster habilitado) (consola). Asegúrese de habilitar el modo de clúster, Cluster Mode enabled (Scale Out) y especifique al menos dos particiones y un nodo de réplica en cada uno.

Creación de un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado) desde cero (AWS CLI)

El siguiente procedimiento crea un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado) mediante la AWS CLI.

Al crear un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado) desde cero, debe crear el grupo de replicación y todos sus nodos con una sola llamada al comando `create-replication-group` de la AWS CLI. Incluya los siguientes parámetros.

--replication-group-id

El nombre del grupo de replicación que va a crear.

Estas son las restricciones de nomenclatura de los grupos de replicación de Valkey o Redis OSS (modo de clúster habilitado):

- Deben contener entre 1 y 40 caracteres alfanuméricos o guiones.
- Deben comenzar por una letra.
- No pueden contener dos guiones consecutivos.
- No pueden terminar con un guion.

--replication-group-description

Descripción del grupo de replicación.

--cache-node-type

El tipo de nodo para cada nodo en el grupo de replicación.

ElastiCache admite los siguientes tipos de nodos. En general, los tipos de la generación actual ofrecen más capacidad de memoria y potencia de computación a menor costo comparados con sus equivalentes de generaciones previas.

Para obtener más información sobre los detalles del rendimiento de cada tipo de nodo, consulte Tipos de instancia de Amazon EC2.

--habilitado para organización de datos en niveles

Establezca este parámetro si está utilizando un tipo de nodo r6gd. Si no quiere la organización de datos en niveles, defina `--no-data-tiering-enabled`. Para obtener más información, consulte Organización de datos en niveles en ElastiCache.

--cache-parameter-group

Especifique el grupo de parámetros `default.redis6.x.cluster.on` o un grupo de parámetros derivado de `default.redis6.x.cluster.on` para crear un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado). Para obtener más información, consulte Cambios de parámetros de Redis OSS 6.x.

--engine

redis

--engine-version

3.2.4

--num-node-groups

El número de grupos de nodo de este grupo de reproducción. Los valores válidos son de 1 a 500.

> **Note**
>
> El límite de nodos o de particiones puede aumentarse hasta un máximo de 500 por clúster. Para solicitar un aumento del límite, consulte AWS Service Limits y seleccione el tipo de límite “Nodes per cluster per instance type” (Nodos por clúster por tipo de instancias).

--replicas-per-node-group

El número de nodos de réplica de cada grupo de nodos. Los valores válidos son de 0 a 5.

--network-type

`ipv4`, `ipv` o `dual-stack`. Si elige la doble pila, debe establecer el parámetro `--IpDiscovery` en `ipv4` o `ipv6`.

Si desea habilitar el cifrado en tránsito o en reposo en este grupo de replicación, agregue uno o estos dos parámetros, `--transit-encryption-enabled` y `--at-rest-encryption-enabled`, y ajústese a las siguientes condiciones.

- El grupo de replicación debe ejecutar la versión de Redis OSS 3.2.6 o 4.0.10.

- El grupo de reproducción debe crearse en una Amazon VPC.
- Debe incluirse también el parámetro `--cache-subnet-group`.
- Debe incluirse el parámetro `--auth-token` con el valor de cadena especificado por el cliente para el token AUTH (contraseña) que es necesario para realizar las operaciones en este grupo de replicación.

La siguiente operación crea el grupo de replicación `sample-repl-group` de Valkey o Redis OSS (modo de clúster habilitado) con tres grupos de nodos/particiones (--num-node-groups), cada uno de ellos con tres nodos, uno principal y dos réplicas de lectura (--replicas-per-node-group).

Para Linux, macOS o Unix:

```
aws elasticache create-replication-group \
   --replication-group-id sample-repl-group \
   --replication-group-description "Demo cluster with replicas" \
   --num-node-groups 3 \
   --replicas-per-node-group 2 \
   --cache-node-type cache.m4.large \
   --engine redis \
   --security-group-ids SECURITY_GROUP_ID \
   --cache-subnet-group-name SUBNET_GROUP_NAME>
```

Para Windows:

```
aws elasticache create-replication-group ^
   --replication-group-id sample-repl-group ^
   --replication-group-description "Demo cluster with replicas" ^
   --num-node-groups 3 ^
   --replicas-per-node-group 2 ^
   --cache-node-type cache.m4.large ^
   --engine redis ^
   --security-group-ids SECURITY_GROUP_ID ^
   --cache-subnet-group-name SUBNET_GROUP_NAME>
```

El comando anterior genera la siguiente salida.

```
{
```

```
    "ReplicationGroup": {
        "Status": "creating",
        "Description": "Demo cluster with replicas",
        "ReplicationGroupId": "sample-repl-group",
        "SnapshotRetentionLimit": 0,
        "AutomaticFailover": "enabled",
        "SnapshotWindow": "05:30-06:30",
        "MemberClusters": [
            "sample-repl-group-0001-001",
            "sample-repl-group-0001-002",
            "sample-repl-group-0001-003",
            "sample-repl-group-0002-001",
            "sample-repl-group-0002-002",
            "sample-repl-group-0002-003",
            "sample-repl-group-0003-001",
            "sample-repl-group-0003-002",
            "sample-repl-group-0003-003"
        ],
        "PendingModifiedValues": {}
    }
}
```

Al crear un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado) desde cero, puede configurar cada partición en el clúster con el parámetro `--node-group-configuration`, tal y como se muestra en el siguiente ejemplo, que configura dos grupos de nodos (consola: particiones). La primera partición tiene dos nodos, un principal y una réplica de lectura. La segunda partición tiene tres nodos, un principal y dos réplicas de lectura.

--node-group-configuration

La configuración para cada grupo de nodos. El parámetro `--node-group-configuration` consta de los siguientes campos.

- `PrimaryAvailabilityZone`: la zona de disponibilidad en la que se encuentra el nodo primario de este grupo de nodos. Si se omite este parámetro, ElastiCache elige la zona de disponibilidad del nodo primario.

  Por ejemplo: us-west-2a.

- `ReplicaAvailabilityZones`: una lista separada por comas de las zonas de disponibilidad en las que se encuentran las réplicas de lectura. El número de zonas de disponibilidad de esta lista debe coincidir con el valor de `ReplicaCount`. Si se omite este parámetro, ElastiCache elige las zonas de disponibilidad de los nodos de réplica.

Ejemplo: "us-west-2a,us-west-2b,us-west-2c"

- `ReplicaCount`: el número de nodos de réplica de este grupo de nodos.
- `Slots`: una cadena que especifica el espacio de claves para el grupo de nodos. La cadena está en formato `startKey-endKey`. Si se omite este parámetro, ElastiCache asigna claves igualmente entre los grupos de nodos.

  Ejemplo: "0-4999"

La siguiente operación crea el grupo de replicación `new-group` de Valkey o Redis OSS (modo de clúster habilitado) con dos grupos de nodos/particiones (`--num-node-groups`). A diferencia del ejemplo anterior, cada grupo de nodos se configura de forma distinta a los demás grupos de nodos (`--node-group-configuration`).

Para Linux, macOS o Unix:

```
aws elasticache create-replication-group \
  --replication-group-id new-group \
  --replication-group-description "Sharded replication group" \
  --engine redis \
  --snapshot-retention-limit 8 \
  --cache-node-type cache.m4.medium \
  --num-node-groups 2 \
  --node-group-configuration \
      "ReplicaCount=1,Slots=0-8999,PrimaryAvailabilityZone='us-
east-1c',ReplicaAvailabilityZones='us-east-1b'" \
      "ReplicaCount=2,Slots=9000-16383,PrimaryAvailabilityZone='us-
east-1a',ReplicaAvailabilityZones='us-east-1a','us-east-1c'"
```

Para Windows:

```
aws elasticache create-replication-group ^
  --replication-group-id new-group ^
  --replication-group-description "Sharded replication group" ^
  --engine redis ^
  --snapshot-retention-limit 8 ^
  --cache-node-type cache.m4.medium ^
  --num-node-groups 2 ^
  --node-group-configuration \
```

```
      "ReplicaCount=1,Slots=0-8999,PrimaryAvailabilityZone='us-
east-1c',ReplicaAvailabilityZones='us-east-1b'" \
      "ReplicaCount=2,Slots=9000-16383,PrimaryAvailabilityZone='us-
east-1a',ReplicaAvailabilityZones='us-east-1a','us-east-1c'"
```

La operación anterior genera la siguiente salida.

```
{
    "ReplicationGroup": {
        "Status": "creating",
        "Description": "Sharded replication group",
        "ReplicationGroupId": "rc-rg",
        "SnapshotRetentionLimit": 8,
        "AutomaticFailover": "enabled",
        "SnapshotWindow": "10:00-11:00",
        "MemberClusters": [
            "rc-rg-0001-001",
            "rc-rg-0001-002",
            "rc-rg-0002-001",
            "rc-rg-0002-002",
            "rc-rg-0002-003"
        ],
        "PendingModifiedValues": {}
    }
}
```

Para obtener información adicional y parámetros que puede querer utilizar, consulte el tema AWS CLI de la create-replication-group.

Creación de un grupo de replicación en Valkey o Redis OSS (modo de clúster habilitado) desde cero (API de ElastiCache)

El siguiente procedimiento crea un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado) mediante la API de ElastiCache.

Al crear un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado) desde cero, debe crear el grupo de replicación y todos sus nodos con una sola llamada a la operación de la API de ElastiCache `CreateReplicationGroup`. Incluya los siguientes parámetros.

ReplicationGroupId

El nombre del grupo de replicación que va a crear.

Estas son las restricciones de nomenclatura de los grupos de replicación de Valkey o Redis OSS (modo de clúster habilitado):

- Deben contener entre 1 y 40 caracteres alfanuméricos o guiones.
- Deben comenzar por una letra.
- No pueden contener dos guiones consecutivos.
- No pueden terminar con un guion.

ReplicationGroupDescription

Descripción del grupo de replicación.

NumNodeGroups

El número de grupos de nodo que desea crear con este grupo de reproducción. Los valores válidos son de 1 a 500.

ReplicasPerNodeGroup

El número de nodos de réplica de cada grupo de nodos. Los valores válidos son de 1 a 5.

NodeGroupConfiguration

La configuración para cada grupo de nodos. El parámetro `NodeGroupConfiguration` consta de los siguientes campos.

- `PrimaryAvailabilityZone`: la zona de disponibilidad en la que se encuentra el nodo primario de este grupo de nodos. Si se omite este parámetro, ElastiCache elige la zona de disponibilidad del nodo primario.

  Por ejemplo: us-west-2a.
- `ReplicaAvailabilityZones`: una lista de las zonas de disponibilidad en las que se encuentran las réplicas de lectura. El número de zonas de disponibilidad de esta lista debe coincidir con el valor de `ReplicaCount`. Si se omite este parámetro, ElastiCache elige las zonas de disponibilidad de los nodos de réplica.
- `ReplicaCount`: el número de nodos de réplica de este grupo de nodos.
- `Slots`: una cadena que especifica el espacio de claves para el grupo de nodos. La cadena está en formato `startKey-endKey`. Si se omite este parámetro, ElastiCache asigna claves igualmente entre los grupos de nodos.

  Ejemplo: "0-4999"

CacheNodeType

El tipo de nodo para cada nodo en el grupo de replicación.

ElastiCache admite los siguientes tipos de nodos. En general, los tipos de la generación actual ofrecen más capacidad de memoria y potencia de computación a menor costo comparados con sus equivalentes de generaciones previas.

Para obtener más información sobre los detalles del rendimiento de cada tipo de nodo, consulte Tipos de instancia de Amazon EC2.

--habilitado para organización de datos en niveles

Establezca este parámetro si está utilizando un tipo de nodo r6gd. Si no quiere la organización de datos en niveles, defina `--no-data-tiering-enabled`. Para obtener más información, consulte Organización de datos en niveles en ElastiCache.

CacheParameterGroup

Especifique el grupo de parámetros `default.redis6.x.cluster.on` o un grupo de parámetros derivado de `default.redis6.x.cluster.on` para crear un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado). Para obtener más información, consulte Cambios de parámetros de Redis OSS 6.x.

--network-type

`ipv4`, `ipv` o `dual-stack`. Si elige la doble pila, debe establecer el parámetro `--IpDiscovery` en `ipv4` o `ipv6`.

Motor

redis

EngineVersion

6.0

Si desea habilitar el cifrado en tránsito o en reposo en este grupo de replicación, agregue uno o estos dos parámetros, `TransitEncryptionEnabled=true` y `AtRestEncryptionEnabled=true`, y ajústese a las siguientes condiciones.

- El grupo de replicación debe ejecutar la versión de Redis OSS 3.2.6 o 4.0.10.
- El grupo de reproducción debe crearse en una Amazon VPC.

- Debe incluirse también el parámetro `CacheSubnetGroup`.
- Debe incluirse el parámetro `AuthToken` con el valor de cadena especificado por el cliente para el token AUTH (contraseña) que es necesario para realizar las operaciones en este grupo de replicación.

Se agregan saltos de línea para facilitar la lectura.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=CreateReplicationGroup
   &CacheNodeType=cache.m4.large
   &CacheParemeterGroup=default.redis6.xcluster.on
   &Engine=redis
   &EngineVersion=6.0
   &NumNodeGroups=3
   &ReplicasPerNodeGroup=2
   &ReplicationGroupDescription=test%20group
   &ReplicationGroupId=myReplGroup
   &Version=2015-02-02
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &X-Amz-Credential=<credential>
```

Para obtener información adicional y parámetros que puede querer utilizar, consulte el tema de la API de ElastiCache CreateReplicationGroup.

Visualización de detalles de un grupo de reproducción

Puede que a veces desee ver los detalles de un grupo de reproducción. Puede utilizar la consola de ElastiCache, la AWS CLI para ElastiCache o la API de ElastiCache. El proceso con la consola es diferente para Valkey o Redis OSS (modo de clúster deshabilitado) que para Valkey o Redis OSS (modo de clúster habilitado).

Visualización de detalles de un grupo de reproducción

- Visualización de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) con réplicas
  - Visualización de los detalles de un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) (consola)
  - Visualización de los detalles de un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) (AWS CLI)

    - Visualización de los detalles de un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) (API de ElastiCache)
- Visualización de un grupo de replicación: Valkey o Redis OSS (modo de clúster deshabilitado)
    - Visualización de un clúster de Valkey o Redis OSS (modo de clúster habilitado) (consola)
    - Visualización de un clúster de Valkey o Redis OSS (modo de clúster habilitado) (AWS CLI)
    - Visualización de un clúster de Valkey o Redis OSS (modo de clúster habilitado) (API de ElastiCache)
- Visualización de detalles de un grupo de reproducción (AWS CLI)
- Visualización de detalles de un grupo de reproducción (API de ElastiCache)

Visualización de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) con réplicas

Puede ver los detalles de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) con réplicas (API/CLI: grupo de replicación) con la consola de ElastiCache, la AWS CLI para ElastiCache o la API de ElastiCache.

Visualización de los detalles de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado)

- Visualización de los detalles de un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) (consola)
- Visualización de los detalles de un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) (AWS CLI)
- Visualización de los detalles de un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) (API de ElastiCache)

Visualización de los detalles de un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) (consola)

Para ver los detalles de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) con réplicas mediante la consola de ElastiCache, consulte el tema Visualización de los detalles de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) (consola).

Visualización de los detalles de un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) (AWS CLI)

Para obtener un ejemplo de la AWS CLI que muestre detalles de un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado), consulte Visualización de detalles de un grupo de reproducción (AWS CLI).

Visualización de los detalles de un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) (API de ElastiCache)

Para obtener un ejemplo de la API de ElastiCache que muestre detalles de un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado), consulte Visualización de detalles de un grupo de reproducción (API de ElastiCache).

Visualización de un grupo de replicación: Valkey o Redis OSS (modo de clúster deshabilitado)

Visualización de un clúster de Valkey o Redis OSS (modo de clúster habilitado) (consola)

Para ver los detalles de un clúster de Valkey o Redis OSS (modo de clúster habilitado) mediante la consola de ElastiCache, consulte Visualización de los detalles de un clúster de Valkey o Redis OSS (modo de clúster habilitado) (consola).

Visualización de un clúster de Valkey o Redis OSS (modo de clúster habilitado) (AWS CLI)

Para obtener un ejemplo de la CLI de ElastiCache que muestre detalles de un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado), consulte Visualización de detalles de un grupo de reproducción (AWS CLI).

Visualización de un clúster de Valkey o Redis OSS (modo de clúster habilitado) (API de ElastiCache)

Para obtener un ejemplo de la API de ElastiCache que muestre detalles de un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado), consulte Visualización de detalles de un grupo de reproducción (API de ElastiCache).

Visualización de detalles de un grupo de reproducción (AWS CLI)

Puede ver los detalles de un grupo de reproducción utilizando el comando AWS CLI de la `describe-replication-groups`. Utilice los siguientes parámetros opcionales para perfeccionar las listas. La omisión de los parámetros devuelve detalles de hasta 100 grupos de reproducción.

Parámetros opcionales

- `--replication-group-id`: utilice este parámetro para enumerar los detalles de un grupo de reproducción específico. En caso de que el grupo de reproducción especificado tenga más de un grupo de nodos, los resultados se devuelven agrupados por grupo de nodos.
- `--max-items`: utilice este parámetro para limitar el número de grupos de reproducción mostrado. El valor de `--max-items` no puede ser inferior a 20 ni superior a 100.

Example

El siguiente código enumera los detalles de hasta 100 grupos de reproducción.

```
aws elasticache describe-replication-groups
```

El siguiente código enumera los detalles de `sample-repl-group`.

```
aws elasticache describe-replication-groups --replication-group-id sample-repl-group
```

El siguiente código enumera los detalles de `sample-repl-group`.

```
aws elasticache describe-replication-groups --replication-group-id sample-repl-group
```

El siguiente código enumera los detalles de hasta 25 grupos de reproducción.

```
aws elasticache describe-replication-groups --max-items 25
```

La salida de esta operación anterior tendrá un aspecto similar al siguiente (formato JSON).

```
{
   "ReplicationGroups": [
     {
       "Status": "available",
       "Description": "test",
       "NodeGroups": [
         {
            "Status": "available",
               "NodeGroupMembers": [
                  {
                     "CurrentRole": "primary",
                     "PreferredAvailabilityZone": "us-west-2a",
```

```
                  "CacheNodeId": "0001",
                  "ReadEndpoint": {
                     "Port": 6379,
                     "Address": "rg-name-001.1abc4d.0001.usw2.cache.amazonaws.com"
                  },
                  "CacheClusterId": "rg-name-001"
               },
               {
                  "CurrentRole": "replica",
                  "PreferredAvailabilityZone": "us-west-2b",
                  "CacheNodeId": "0001",
                  "ReadEndpoint": {
                     "Port": 6379,
                     "Address": "rg-name-002.1abc4d.0001.usw2.cache.amazonaws.com"
                  },
                  "CacheClusterId": "rg-name-002"
               },
               {
                  "CurrentRole": "replica",
                  "PreferredAvailabilityZone": "us-west-2c",
                  "CacheNodeId": "0001",
                  "ReadEndpoint": {
                     "Port": 6379,
                     "Address": "rg-name-003.1abc4d.0001.usw2.cache.amazonaws.com"
                  },
                  "CacheClusterId": "rg-name-003"
               }
            ],
            "NodeGroupId": "0001",
            "PrimaryEndpoint": {
               "Port": 6379,
               "Address": "rg-name.1abc4d.ng.0001.usw2.cache.amazonaws.com"
            }
         }
      ],
      "ReplicationGroupId": "rg-name",
      "AutomaticFailover": "enabled",
      "SnapshottingClusterId": "rg-name-002",
      "MemberClusters": [
         "rg-name-001",
         "rg-name-002",
         "rg-name-003"
      ],
      "PendingModifiedValues": {}
```

```
      },
      {
      ... some output omitted for brevity
      }
   ]
}
```

Para obtener más información, consulte la AWS CLI para el tema de ElastiCache describe-replication-groups.

Visualización de detalles de un grupo de reproducción (API de ElastiCache)

Puede ver los detalles de una reproducción utilizando la operación AWS CLI de la `DescribeReplicationGroups`. Utilice los siguientes parámetros opcionales para perfeccionar las listas. La omisión de los parámetros devuelve detalles de hasta 100 grupos de reproducción.

Parámetros opcionales

- `ReplicationGroupId`: utilice este parámetro para enumerar los detalles de un grupo de reproducción específico. En caso de que el grupo de reproducción especificado tenga más de un grupo de nodos, los resultados se devuelven agrupados por grupo de nodos.
- `MaxRecords`: utilice este parámetro para limitar el número de grupos de reproducción mostrado. El valor de `MaxRecords` no puede ser inferior a 20 ni superior a 100. El valor predeterminado es 100.

Example

El siguiente código enumera los detalles de hasta 100 grupos de reproducción.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DescribeReplicationGroups
   &Version=2015-02-02
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &X-Amz-Credential=<credential>
```

El siguiente código enumera los detalles de `myReplGroup`.

```
https://elasticache.us-west-2.amazonaws.com/
```

```
   ?Action=DescribeReplicationGroups
   &ReplicationGroupId=myReplGroup
   &Version=2015-02-02
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &X-Amz-Credential=<credential>
```

El siguiente código enumera los detalles de hasta 25 clústeres.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DescribeReplicationGroups
   &MaxRecords=25
   &Version=2015-02-02
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &X-Amz-Credential=<credential>
```

Para obtener más información, consulte el tema de referencia de la API de ElastiCache DescribeReplicationGroups.

Búsqueda de puntos de enlace del grupo de reproducción

Una aplicación puede conectarse a cualquier nodo en un grupo de reproducción, siempre que tenga el punto de conexión de DNS y el número de puerto de ese nodo. En función de si ejecuta un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) o de Valkey o Redis OSS (modo de clúster habilitado), le interesarán distintos puntos de conexión.

Valkey o Redis OSS (modo de clúster deshabilitado)

Los clústeres de Valkey o Redis OSS (modo de clúster deshabilitado) con réplicas tienen tres tipos de puntos de conexión: el punto de conexión principal, el punto de conexión del lector y los puntos de conexión de nodos. El punto de conexión principal es un nombre de DNS que siempre se resuelve en el nodo principal del clúster. El punto de conexión principal es inmune a los cambios en el clúster, como la promoción de una réplica de lectura a la función principal. Para una actividad de escritura, recomendamos que sus aplicaciones se conecten al punto de conexión principal.

Un punto de conexión del lector dividirá de manera uniforme las conexiones entrantes al punto de conexión entre todas las réplicas de lectura de un clúster de ElastiCache. Factores adicionales como cuando la aplicación crea las conexiones o cómo la aplicación (re)utiliza las conexiones determinará la distribución del tráfico. Los puntos de enlace de lector se mantienen actualizados con los cambios del clúster en tiempo real cuando se añaden o eliminan réplicas. Puede colocar las diferentes réplicas de lectura del clúster de ElastiCache para Redis OSS en distintas zonas de disponibilidad (AZ) de AWS para garantizar la alta disponibilidad de los puntos de conexión del lector.

> **Note**
>
> Un punto de conexión del lector no es un equilibrador de carga. Se trata de un registro de DNS que se resolverá en una dirección IP de uno de los nodos de réplica en forma de turno rotativo.

Para la actividad de lectura, las aplicaciones pueden conectarse también a cualquier nodo del clúster. A diferencia del punto de conexión principal, los puntos de conexión de nodo se resuelven en puntos de enlace específicos. Si realiza un cambio en su clúster, como añadir o eliminar una réplica, debe actualizar los puntos de enlace del nodo en su aplicación.

Valkey o Redis OSS (modo de clúster habilitado)

Los clústeres de Valkey o Redis OSS (modo de clúster habilitado) con réplicas, debido a que cuentan con varias particiones (API/CLI: grupos de nodos), lo que significa que también tienen varios nodos

principales, cuentan con una estructura de puntos de conexión distinta a la de los clústeres de Valkey o Redis OSS (modo de clúster deshabilitado). Valkey o Redis OSS (modo de clúster habilitado) tienen un punto de conexión de configuración que conoce todos los puntos de conexión principales y de nodo del clúster. La aplicación se conecta al punto de conexión de configuración. Cuando su aplicación escribe o lee en el punto de conexión de configuración del clúster, Valkey o Redis OSS determinan, en segundo plano, a qué partición pertenece la clave y qué punto de conexión de dicha partición se debe utilizar. Todo es bastante transparente para su aplicación.

Puede buscar los puntos de enlace de un clúster mediante la consola de ElastiCache, la AWS CLI o la API de ElastiCache.

Búsqueda de puntos de enlace del grupo de reproducción

Para encontrar los puntos de enlace de su grupo de reproducción, consulte uno de los siguientes temas:

- Búsqueda de puntos de conexión de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) (consola)
- Búsqueda de puntos de conexión para un clúster de Valkey o Redis OSS (modo de clúster habilitado) (consola)
- Búsqueda de los puntos de conexión de los grupos de replicación de Valkey o Redis OSS (AWS CLI)
- Búsqueda de puntos de conexión de grupos de replicación de Valkey o Redis OSS (API de ElastiCache)

Modificación de un grupo de reproducción

⚠ Restricciones importantes

- Actualmente, ElastiCache admite modificaciones limitadas de un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado), por ejemplo, cambiar la versión del motor, mediante la operación de la API `ModifyReplicationGroup` (CLI: `modify-replication-group`). Puede modificar el número de particiones (grupos de nodos) en un clúster de Valkey o Redis OSS (modo de clúster habilitado) con la operación `ModifyReplicationGroupShardConfiguration` de la API (CLI: `modify-replication-group-shard-configuration`). Para obtener más información, consulte Escalado de clústeres de Valkey o Redis OSS (modo de clúster habilitado).

  Otras modificaciones en un clúster de Valkey o Redis OSS (modo de clúster habilitado) requieren que cree un clúster con el clúster nuevo y los cambios incorporados.
- Puede actualizar clústeres y grupos de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) y Valkey o Redis OSS (modo de clúster habilitado) a versiones del motor más recientes. Sin embargo, no se puede volver a versiones del motor anteriores, salvo si se elimina el clúster o el motor de reproducción existente y se vuelve a crear. Para obtener más información, consulte Administración de versiones para ElastiCache.
- Puede actualizar un clúster de ElastiCache para Valkey o Redis OSS existente que utilice el modo de clúster deshabilitado para usar el modo de clúster habilitado mediante la consola, la API ModifyReplicationGroup o el comando de la CLI modify-replication-group, tal como se muestra en el siguiente ejemplo. También puede seguir los pasos que se indican en Modificación del modo de clúster.

Puede modificar la configuración de un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) con la consola de ElastiCache, la AWS CLI o la API de ElastiCache. Actualmente, ElastiCache admite un número limitado de modificaciones en un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado). Otras modificaciones requieren crear una copia de seguridad del grupo de replicación actual y, a continuación, utilizar dicha copia para inicializar un grupo de replicación nuevo de Valkey o Redis OSS (modo de clúster habilitado).

Temas

- Uso de Consola de administración de AWS

- Uso de AWS CLI
- Uso de la API de ElastiCache

Uso de Consola de administración de AWS

Para modificar un clúster de Valkey o Redis OSS (modo de clúster deshabilitado), consulte Modificación de un clúster de ElastiCache.

Uso de AWS CLI

A continuación, se muestran ejemplos de la AWS CLI del comando `modify-replication-group`. Puede utilizar el mismo comando para realizar otras modificaciones en un grupo de reproducción.

Habilitación de multi-AZ en un grupo de replicación de Valkey o Redis OSS existente:

Para Linux, macOS o Unix:

```
aws elasticache modify-replication-group \
   --replication-group-id myReplGroup \
   --multi-az-enabled = true
```

Para Windows:

```
aws elasticache modify-replication-group ^
   --replication-group-id myReplGroup ^
   --multi-az-enabled
```

Modifique el modo de clúster de desactivado a habilitado:

Para modificar el modo de clúster de desactivado a habilitado, primero debe establecer el modo de clúster en compatible. El modo compatible permite que los clientes de Valkey o Redis OSS se conecten mediante el modo clúster habilitado y el modo clúster desactivado. Después de migrar todos los clientes de Valkey o Redis OSS para usar el modo de clúster habilitado, puede completar la configuración del modo de clúster y establecer el modo de clúster en Habilitado.

Para Linux, macOS o Unix:

Establezca el modo de clúster en compatible.

```
aws elasticache modify-replication-group \
```

```
   --replication-group-id myReplGroup \
   --cache-parameter-group-name myParameterGroupName \
   --cluster-mode compatible
```

Establezca el modo de clúster en habilitado.

```
aws elasticache modify-replication-group \
   --replication-group-id myReplGroup \
   --cluster-mode enabled
```

Para Windows:

Establezca el modo de clúster en compatible.

```
aws elasticache modify-replication-group ^
   --replication-group-id myReplGroup ^
   --cache-parameter-group-name myParameterGroupName ^
   --cluster-mode compatible
```

Establezca el modo de clúster en habilitado.

```
aws elasticache modify-replication-group ^
   --replication-group-id myReplGroup ^
   --cluster-mode enabled
```

Para obtener más información sobre el comando `modify-replication-group` de la AWS CLI, consulte modify-replication-group o Modificación del modo de clúster en la Guía del usuario de ElastiCache para Redis OSS.

Uso de la API de ElastiCache

La siguiente operación de la API de ElastiCache habilita multi-AZ en un grupo de replicación existente de Valkey o Redis OSS. Puede utilizar la misma operación para realizar otras modificaciones en un grupo de reproducción.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=ModifyReplicationGroup
   &AutomaticFailoverEnabled=true
   &Mutli-AZEnabled=true
   &ReplicationGroupId=myReplGroup
```

```
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &Version=2014-12-01
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20141201T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Para obtener más información sobre la operación `ModifyReplicationGroup` de la API de ElastiCache, consulte ModifyReplicationGroup.

Eliminación de un grupo de reproducción

Si ya no necesita uno de los clústeres con réplicas (denominados grupos de reproducción en la API/CLI), puede eliminarlo. Al eliminar un grupo de reproducción, ElastiCache elimina todos los nodos de ese grupo.

Una vez que haya iniciado esta operación, no se puede interrumpir ni cancelar.

> **Warning**
>
> - Cuando se elimina un clúster de ElastiCache para Redis OSS, se retienen las instantáneas manuales. También tendrá la opción de crear una instantánea final antes de la eliminación del clúster. Por el contrario, las instantáneas de caché automáticas no se conservan.
> - El permiso `CreateSnapshot` es necesario para crear una instantánea final. Sin este permiso, la llamada a la API fallará con una excepción `Access Denied`.

Eliminación de un grupo de reproducción (consola)

Para eliminar un clúster que tiene réplicas, consulte Eliminación de un clúster en ElastiCache.

Eliminación de un grupo de reproducción de (AWS CLI)

Use el comando delete-replication-group para eliminar un grupo de reproducción.

```
aws elasticache delete-replication-group --replication-group-id my-repgroup
```

Aparecerá una pregunta que pedirá que confirme su decisión. Escriba y (sí) para iniciar la operación de forma inmediata. Después de que comience el proceso, este será irreversible.

```
  After you begin deleting this replication group, all of its nodes will be deleted as
 well.
  Are you sure you want to delete this replication group? [Ny]y

REPLICATIONGROUP  my-repgroup  My replication group  deleting
```

Eliminación de un grupo de reproducción (API de ElastiCache)

Llame a DeleteReplicationGroup con el parámetro `ReplicationGroup`.

Example

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DeleteReplicationGroup
   &ReplicationGroupId=my-repgroup
   &Version=2014-12-01
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20141201T220302Z
   &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
   &X-Amz-Date=20141201T220302Z
   &X-Amz-SignedHeaders=Host
   &X-Amz-Expires=20141201T220302Z
   &X-Amz-Credential=<credential>
   &X-Amz-Signature=<signature>
```

> **Note**
>
> Si establece el parámetro `RetainPrimaryCluster` en `true`, todas las réplicas de lectura se eliminarán, pero el clúster principal se conservará.

Cambio del número de réplicas

Puede aumentar o disminuir de forma dinámica el número de réplicas de lectura del grupo de replicación de Valkey o Redis OSS mediante la Consola de administración de AWS, la AWS CLI o la API de ElastiCache. Si el grupo de replicación es un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado), puede elegir en qué particiones (grupos de nodos) va a aumentar o disminuir el número de réplicas.

Para cambiar dinámicamente el número de réplicas del grupo de replicación, elija en la tabla siguiente la operación que mejor se ajuste a sus necesidades.

Para hacer esto	Para Valkey o Redis OSS (modo de clúster habilitado)	Para Valkey o Redis OSS (modo de clúster deshabilitado)
Añadir réplicas	Aumento del número de réplicas de una partición	Aumento del número de réplicas de una partición Adición de una réplica de lectura para Valkey o Redis OSS (modo de clúster deshabilitado)
Eliminar réplicas	Reducción del número de réplicas de una partición	Reducción del número de réplicas de una partición Eliminación de una réplica de lectura para Valkey o Redis OSS (modo de clúster deshabilitado)

Aumento del número de réplicas de una partición

Puede aumentar el número de réplicas en una partición de Valkey o Redis OSS (modo de clúster habilitado) o en un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) hasta un máximo de cinco. Para ello, utilice la Consola de administración de AWS, la AWS CLI o la API de ElastiCache.

Temas

- Uso de Consola de administración de AWS
- Uso de AWS CLI
- Uso de la API de ElastiCache

Uso de Consola de administración de AWS

El siguiente procedimiento utiliza la consola para aumentar el número de réplicas de un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado).

Aumento del número de réplicas en las particiones

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Valkey o Redis OSS y, a continuación, elija el nombre del grupo de replicación al que desea añadir las réplicas.
3. Elija el cuadro correspondiente a cada fragmento al que desea agregar réplicas.
4. Elija Add replicas (Agregar réplicas).
5. Complete la página Add Replicas to Shards (Agregar réplicas a las particiones):
    - En New number of replicas/shard (Nuevo número de réplicas por partición), introduzca el número de réplicas que desea que tengan todos los fragmentos seleccionados. Este valor debe ser mayor o igual que Current Number of Replicas per shard (Número actual de réplicas por fragmento) y menor o igual que cinco. Recomendamos dos réplicas como mínimo.
    - En Availability Zones (Zonas de disponibilidad), elija No preference (Sin preferencia) a fin de que ElastiCache elija una zona de disponibilidad para cada réplica nueva, o Specify Availability Zones (Especificar las zonas de disponibilidad) si desea elegir una zona de disponibilidad para cada réplica nueva.

Si elige Specify Availability Zones (Especificar las zonas de disponibilidad), utilice la lista para especificar una zona de disponibilidad para cada réplica nueva.

6. Elija Add (Agregar) para agregar las réplicas o Cancel (Cancelar) para cancelar la operación.

Uso de AWS CLI

Para aumentar el número de réplicas de una partición de Valkey o Redis OSS, utilice el comando `increase-replica-count` con los parámetros siguientes:

- `--replication-group-id`: obligatorio. Identifica el grupo de replicación en el que desea aumentar el número de réplicas.
- `--apply-immediately` o `--no-apply-immediately`: obligatorios. Especifica si se debe aumentar el número de réplicas inmediatamente (`--apply-immediately`) o en el periodo de mantenimiento siguiente (`--no-apply-immediately`). Actualmente, no se admite `--no-apply-immediately`.
- `--new-replica-count`: opcional. Especifica el número de nodos de réplica que se desea al terminar, hasta un máximo de cinco. Utilice este parámetro en los grupos de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) en los que solo haya un grupo de nodos o un grupo de Valkey o Redis OSS (modo de clúster habilitado), o en aquellos en los que desee que todos los grupos de nodos tengan el mismo número de réplicas. Si este valor no es superior al número actual de réplicas del grupo de nodos, la llamada produce una excepción.
- `--replica-configuration`: opcional. Le permite establecer el número de réplicas y de zonas de disponibilidad para cada grupo de nodos de forma independiente. Utilice este parámetro en los grupos de Valkey o Redis OSS (modo de clúster habilitado) en los que desee configurar cada grupo de nodos por separado.

  `--replica-configuration` tiene tres miembros opcionales:

  - `NodeGroupId`: el ID de cuatro dígitos del grupo de nodos que está configurando. El ID de partición de los grupos de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) es siempre `0001`. Para buscar un ID de un grupo de nodos (particiones) de Valkey o Redis OSS (modo de clúster habilitado), consulte Búsqueda del ID de una partición.
  - `NewReplicaCount`: el número de réplicas que desea incluir en este grupo de nodos al final de la operación. El valor debe ser superior al número actual de réplicas, hasta un máximo de cinco. Si este valor no es superior al número actual de réplicas del grupo de nodos, la llamada produce una excepción.

- `PreferredAvailabilityZones`: una lista de cadenas `PreferredAvailabilityZone` que especifica en qué zonas de disponibilidad se encuentran los nodos del grupo de reproducción. El número de valores `PreferredAvailabilityZone` debe ser igual al valor de `NewReplicaCount` más 1 para incluir el nodo principal. Si se omite este miembro de `--replica-configuration`, ElastiCache para Redis OSS elegirá la zona de disponibilidad para cada una de las réplicas nuevas.

> **Important**
>
> En la llamada, debe incluir el parámetro `--new-replica-count` o `--replica-configuration`, pero no ambos.

Example

En el siguiente ejemplo, se aumenta el número de réplicas del grupo de reproducción `sample-repl-group` a tres. Al finalizar el ejemplo, existirán tres réplicas en cada grupo de nodos. Este número se aplica tanto si este es un grupo de Valkey o Redis OSS (modo de clúster deshabilitado) con un único grupo de nodos, como si es un grupo de Valkey o Redis OSS (modo de clúster habilitado) con varios grupos de nodos.

Para Linux, macOS o Unix:

```
aws elasticache increase-replica-count \
    --replication-group-id sample-repl-group \
    --new-replica-count 3 \
    --apply-immediately
```

Para Windows:

```
aws elasticache increase-replica-count ^
    --replication-group-id sample-repl-group ^
    --new-replica-count 3 ^
    --apply-immediately
```

En el siguiente ejemplo, se aumenta el número de réplicas del grupo de replicación `sample-repl-group` al valor indicado para los dos grupos de nodos especificados. Dado que hay varios grupos de nodos, este es un grupo de replicación de Valkey o Redis OSS (modo de clúster

habilitado). Al especificar el valor opcional `PreferredAvailabilityZones`, el número de zonas de disponibilidad indicadas debe ser igual al valor de `NewReplicaCount` más 1. Este enfoque incluye el nodo principal del grupo identificado por `NodeGroupId`.

Para Linux, macOS o Unix:

```
aws elasticache increase-replica-count \
    --replication-group-id sample-repl-group \
    --replica-configuration \
        NodeGroupId=0001,NewReplicaCount=2,PreferredAvailabilityZones=us-east-1a,us-east-1c,us-east-1b \
        NodeGroupId=0003,NewReplicaCount=3,PreferredAvailabilityZones=us-east-1a,us-east-1b,us-east-1c,us-east-1c \
    --apply-immediately
```

Para Windows:

```
aws elasticache increase-replica-count ^
    --replication-group-id sample-repl-group ^
    --replica-configuration ^
        NodeGroupId=0001,NewReplicaCount=2,PreferredAvailabilityZones=us-east-1a,us-east-1c,us-east-1b ^
        NodeGroupId=0003,NewReplicaCount=3,PreferredAvailabilityZones=us-east-1a,us-east-1b,us-east-1c,us-east-1c \
    --apply-immediately
```

Para obtener más información sobre cómo aumentar el número de réplicas mediante la CLI, consulte increase-replica-count en la Referencia de línea de comandos de Amazon ElastiCache.

Uso de la API de ElastiCache

Para aumentar el número de réplicas de una partición de Valkey o Redis OSS, utilice la acción `IncreaseReplicaCount` con los parámetros siguientes:

- `ReplicationGroupId`: obligatorio. Identifica el grupo de replicación en el que desea aumentar el número de réplicas.
- `ApplyImmediately`: obligatorio. Especifica si se debe aumentar el número de réplicas inmediatamente (`ApplyImmediately=True`) o en el periodo de mantenimiento siguiente (`ApplyImmediately=False`). Actualmente, no se admite `ApplyImmediately=False`.
- `NewReplicaCount`: opcional. Especifica el número de nodos de réplica que se desea al terminar, hasta un máximo de cinco. Utilice este parámetro en los grupos de replicación de Valkey o Redis

OSS (modo de clúster deshabilitado) en los que solo haya un grupo de nodos, o bien grupos de Valkey o Redis OSS (modo de clúster habilitado) en los que desee que todos los grupos de nodos tengan el mismo número de réplicas. Si este valor no es superior al número actual de réplicas del grupo de nodos, la llamada produce una excepción.

- `ReplicaConfiguration`: opcional. Le permite establecer el número de réplicas y de zonas de disponibilidad para cada grupo de nodos de forma independiente. Utilice este parámetro en los grupos de Valkey o Redis OSS (modo de clúster habilitado) en los que desee configurar cada grupo de nodos por separado.

  `ReplicaConfiguraion` tiene tres miembros opcionales:

  - `NodeGroupId`: el ID de cuatro dígitos del grupo de nodos que está configurando. Para grupos de replicación de Valkey o Redis OSS (modo de clúster deshabilitado), el ID del grupo de nodos (partición) siempre es `0001`. Para buscar un ID de un grupo de nodos (particiones) de Valkey o Redis OSS (modo de clúster habilitado), consulte Búsqueda del ID de una partición.
  - `NewReplicaCount`: el número de réplicas que desea incluir en este grupo de nodos al final de la operación. El valor debe ser superior al número actual de réplicas, hasta un máximo de cinco. Si este valor no es superior al número actual de réplicas del grupo de nodos, la llamada produce una excepción.
  - `PreferredAvailabilityZones`: una lista de cadenas `PreferredAvailabilityZone` que especifica en qué zonas de disponibilidad se encuentran los nodos del grupo de reproducción. El número de valores `PreferredAvailabilityZone` debe ser igual al valor de `NewReplicaCount` más 1 para incluir el nodo principal. Si se omite este miembro de `ReplicaConfiguration`, ElastiCache para Redis OSS elegirá la zona de disponibilidad para cada una de las réplicas nuevas.

> **Important**
>
> En la llamada, debe incluir el parámetro `NewReplicaCount` o `ReplicaConfiguration`, pero no ambos.

Example

En el siguiente ejemplo, se aumenta el número de réplicas del grupo de reproducción `sample-repl-group` a tres. Al finalizar el ejemplo, existirán tres réplicas en cada grupo de nodos. Este número se aplica tanto si este es un grupo de Valkey o Redis OSS (modo de clúster deshabilitado)

con un único grupo de nodos, como si es un grupo de Valkey o Redis OSS (modo de clúster habilitado) con varios grupos de nodos.

```
https://elasticache.us-west-2.amazonaws.com/
      ?Action=IncreaseReplicaCount
      &ApplyImmediately=True
      &NewReplicaCount=3
      &ReplicationGroupId=sample-repl-group
      &Version=2015-02-02
      &SignatureVersion=4
      &SignatureMethod=HmacSHA256
      &Timestamp=20150202T192317Z
      &X-Amz-Credential=<credential>
```

En el siguiente ejemplo, se aumenta el número de réplicas del grupo de replicación `sample-repl-group` al valor indicado para los dos grupos de nodos especificados. Dado que hay varios grupos de nodos, este es un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado). Al especificar el valor opcional `PreferredAvailabilityZones`, el número de zonas de disponibilidad indicadas debe ser igual al valor de `NewReplicaCount` más 1. Este enfoque incluye el nodo principal del grupo identificado por `NodeGroupId`.

```
https://elasticache.us-west-2.amazonaws.com/
      ?Action=IncreaseReplicaCount
      &ApplyImmediately=True
      &ReplicaConfiguration.ConfigureShard.1.NodeGroupId=0001
      &ReplicaConfiguration.ConfigureShard.1.NewReplicaCount=2

 &ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.1=
east-1a

 &ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.2=
east-1c

 &ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.3=
east-1b
      &ReplicaConfiguration.ConfigureShard.2.NodeGroupId=0003
      &ReplicaConfiguration.ConfigureShard.2.NewReplicaCount=3

 &ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.1=
east-1a
```

```
 &ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.2=
east-1b

 &ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.3=
east-1c

 &ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.4=
east-1c
      &ReplicationGroupId=sample-repl-group
      &Version=2015-02-02
      &SignatureVersion=4
      &SignatureMethod=HmacSHA256
      &Timestamp=20150202T192317Z
      &X-Amz-Credential=<credential>
```

Para obtener más información sobre cómo aumentar el número de réplicas mediante la API, consulte IncreaseReplicaCount en la Referencia de la API de Amazon ElastiCache.

Reducción del número de réplicas de una partición

Puede reducir el número de réplicas de una partición de Valkey o Redis OSS (modo de clúster habilitado) o en un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado):

- En Valkey o Redis OSS (modo de clúster deshabilitado), puede reducir el número de réplicas a una si multi-AZ se encuentra habilitado, y a cero, si multi-AZ no se encuentra habilitado.
- En Valkey o Redis OSS (modo de clúster habilitado), puede reducir el número de réplicas a cero. Sin embargo, no es posible realizar la conmutación por error a una réplica si se produce un error en el nodo principal.

Puede utilizar la Consola de administración de AWS, la AWS CLI o la API de ElastiCache para reducir el número de réplicas de un grupo de nodos (partición) o un grupo de reproducción.

Temas

- Uso de Consola de administración de AWS
- Uso de AWS CLI
- Uso de la API de ElastiCache

Uso de Consola de administración de AWS

El siguiente procedimiento utiliza la consola para reducir el número de réplicas de un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado).

Reducción del número de réplicas en una partición de Valkey o Redis OSS

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Valkey o Redis OSS y, a continuación, elija el nombre del grupo de replicación del que desea eliminar las réplicas.
3. Elija el cuadro correspondiente a cada fragmento del que desea eliminar un nodo de réplica.
4. Elija Delete replicas (Eliminar réplicas).
5. Complete la página Delete Replicas from Shards (Eliminar réplicas de particiones):

    a. En New number of replicas/shard (Nuevo número de réplicas por partición), introduzca el número de réplicas que desea que tengan los fragmentos seleccionados. Este número debe ser mayor o igual que 1. Recomendamos dos réplicas por fragmento como mínimo.

b. Elija Delete (Eliminar) para eliminar las réplicas o Cancel (Cancelar) para cancelar la operación.

> **Important**
>
> - Si no especifica los nodos de réplica que se deben eliminar, ElastiCache para Redis OSS los seleccionará de forma automática para eliminarlos. Al hacerlo, ElastiCache para Redis OSS intenta retener la arquitectura de Multi-AZ para el grupo de replicación y mantener las réplicas con un mínimo retraso de replicación con el nodo principal.
> - No puede eliminar los nodos primarios o el nodo primario de un grupo de reproducción. Si especifica la eliminación del nodo principal, la operación producirá un error con un evento que indica que el nodo principal se ha seleccionado para su eliminación.

Uso de AWS CLI

Para reducir el número de réplicas de una partición de Valkey o Redis OSS, utilice el comando `decrease-replica-count` con los parámetros siguientes:

- `--replication-group-id`: obligatorio. Identifica el grupo de replicación en el que se desea reducir el número de réplicas.
- `--apply-immediately` o `--no-apply-immediately`: obligatorios. Especifica si se debe reducir el número de réplicas inmediatamente (`--apply-immediately`) o en el periodo de mantenimiento siguiente (`--no-apply-immediately`). Actualmente, no se admite `--no-apply-immediately`.
- `--new-replica-count`: opcional. Especifica el número de nodos de réplica que se desean. El valor de `--new-replica-count` debe ser un valor válido inferior al número actual de réplicas de los grupos de nodos. Para conocer los valores mínimos permitidos, consulte Reducción del número de réplicas de una partición. Si el valor de `--new-replica-count` no cumple este requisito, la llamada produce un error.
- `--replicas-to-remove`: opcional. Contiene una lista de los ID de nodo que especifican los nodos de réplica que se van a eliminar.
- `--replica-configuration`: opcional. Le permite establecer el número de réplicas y de zonas de disponibilidad para cada grupo de nodos de forma independiente. Utilice este parámetro en

los grupos de Valkey o Redis OSS (modo de clúster habilitado) en los que desee configurar cada grupo de nodos por separado.

`--replica-configuration` tiene tres miembros opcionales:

- `NodeGroupId`: el ID de cuatro dígitos del grupo de nodos que está configurando. El ID de partición de los grupos de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) es siempre `0001`. Para buscar un ID de un grupo de nodos (particiones) de Valkey o Redis OSS (modo de clúster habilitado), consulte Búsqueda del ID de una partición.
- `NewReplicaCount`: un parámetro opcional que especifica el número de nodos de réplica que desea. El valor de `NewReplicaCount` debe ser un valor válido inferior al número actual de réplicas de los grupos de nodos. Para conocer los valores mínimos permitidos, consulte Reducción del número de réplicas de una partición. Si el valor de `NewReplicaCount` no cumple este requisito, la llamada produce un error.
- `PreferredAvailabilityZones`: una lista de cadenas `PreferredAvailabilityZone` que especifica en qué zonas de disponibilidad se encuentran los nodos del grupo de reproducción. El número de valores `PreferredAvailabilityZone` debe ser igual al valor de `NewReplicaCount` más 1 para incluir el nodo principal. Si se omite este miembro de `--replica-configuration`, ElastiCache para Redis OSS elegirá la zona de disponibilidad para cada una de las réplicas nuevas.

> **Important**
>
> Debe incluir únicamente uno de los parámetros `--new-replica-count`, `--replicas-to-remove` o `--replica-configuration`.

Example

En el siguiente ejemplo, se utiliza `--new-replica-count` para reducir el número de réplicas del grupo de reproducción `sample-repl-group` a una. Al finalizar el ejemplo, existirá una réplica en cada grupo de nodos. Este número se aplica tanto si este es un grupo de Valkey o Redis OSS (modo de clúster deshabilitado) con un único grupo de nodos, como si es un grupo de Valkey o Redis OSS (modo de clúster habilitado) con varios grupos de nodos.

Para Linux, macOS o Unix:

```
aws elasticache decrease-replica-count
```

```
    --replication-group-id sample-repl-group \
    --new-replica-count 1 \
    --apply-immediately
```

Para Windows:

```
aws elasticache decrease-replica-count ^
    --replication-group-id sample-repl-group ^
    --new-replica-count 1 ^
    --apply-immediately
```

En el siguiente ejemplo, se reduce el número de réplicas del grupo de reproducción `sample-repl-group` eliminando dos réplicas especificadas (`0001` y `0003`) del grupo de nodos.

Para Linux, macOS o Unix:

```
aws elasticache decrease-replica-count \
    --replication-group-id sample-repl-group \
    --replicas-to-remove 0001,0003 \
    --apply-immediately
```

Para Windows:

```
aws elasticache decrease-replica-count ^
    --replication-group-id sample-repl-group ^
    --replicas-to-remove 0001,0003 \
    --apply-immediately
```

En el siguiente ejemplo, se utiliza `--replica-configuration` para reducir el número de réplicas del grupo de reproducción `sample-repl-group` al valor indicado para los dos grupos de nodos especificados. Dado que hay varios grupos de nodos, este es un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado). Al especificar el valor opcional `PreferredAvailabilityZones`, el número de zonas de disponibilidad indicadas debe ser igual al valor de `NewReplicaCount` más 1. Este enfoque incluye el nodo principal del grupo identificado por `NodeGroupId`.

Para Linux, macOS o Unix:

```
aws elasticache decrease-replica-count \
```

```
    --replication-group-id sample-repl-group \
    --replica-configuration \
        NodeGroupId=0001,NewReplicaCount=1,PreferredAvailabilityZones=us-east-1a,us-
east-1c \
        NodeGroupId=0003,NewReplicaCount=2,PreferredAvailabilityZones=us-east-1a,us-
east-1b,us-east-1c \
    --apply-immediately
```

Para Windows:

```
aws elasticache decrease-replica-count ^
    --replication-group-id sample-repl-group ^
    --replica-configuration ^
        NodeGroupId=0001,NewReplicaCount=2,PreferredAvailabilityZones=us-east-1a,us-
east-1c ^
        NodeGroupId=0003,NewReplicaCount=3,PreferredAvailabilityZones=us-east-1a,us-
east-1b,us-east-1c \
    --apply-immediately
```

Para obtener más información sobre cómo reducir el número de réplicas mediante la CLI, consulte decrease-replica-count en la Referencia de línea de comandos de Amazon ElastiCache.

Uso de la API de ElastiCache

Para reducir el número de réplicas de una partición de Valkey o Redis OSS, utilice la acción `DecreaseReplicaCount` con los parámetros siguientes:

- `ReplicationGroupId`: obligatorio. Identifica el grupo de replicación en el que se desea reducir el número de réplicas.
- `ApplyImmediately`: obligatorio. Especifica si se debe reducir el número de réplicas inmediatamente (`ApplyImmediately=True`) o en el periodo de mantenimiento siguiente (`ApplyImmediately=False`). Actualmente, no se admite `ApplyImmediately=False`.
- `NewReplicaCount`: opcional. Especifica el número de nodos de réplica que se desean. El valor de `NewReplicaCount` debe ser un valor válido inferior al número actual de réplicas de los grupos de nodos. Para conocer los valores mínimos permitidos, consulte Reducción del número de réplicas de una partición. Si el valor de `--new-replica-count` no cumple este requisito, la llamada produce un error.
- `ReplicasToRemove`: opcional. Contiene una lista de los ID de nodo que especifican los nodos de réplica que se van a eliminar.

- `ReplicaConfiguration`: opcional. Contiene una lista de grupos de nodos que permite establecer el número de réplicas y de zonas de disponibilidad para cada grupo de nodos de forma independiente. Utilice este parámetro en los grupos de Valkey o Redis OSS (modo de clúster habilitado) en los que desee configurar cada grupo de nodos por separado.

  `ReplicaConfiguraion` tiene tres miembros opcionales:

  - `NodeGroupId`: el ID de cuatro dígitos del grupo de nodos que está configurando. Para grupos de replicación de Valkey o Redis OSS (modo de clúster deshabilitado), el ID del grupo de nodos siempre es `0001`. Para buscar un ID de un grupo de nodos (particiones) de Valkey o Redis OSS (modo de clúster habilitado), consulte Búsqueda del ID de una partición.
  - `NewReplicaCount`: el número de réplicas que desea incluir en este grupo de nodos al final de la operación. El valor debe ser inferior al número actual de réplicas hasta un mínimo de 1 si se ha habilitado Multi-AZ o de 0 si Multi-AZ con conmutación por error automática no está habilitado. Si este valor no es menor que el número actual de réplicas del grupo de nodos, la llamada produce una excepción.
  - `PreferredAvailabilityZones`: una lista de cadenas `PreferredAvailabilityZone` que especifica en qué zonas de disponibilidad se encuentran los nodos del grupo de reproducción. El número de valores `PreferredAvailabilityZone` debe ser igual al valor de `NewReplicaCount` más 1 para incluir el nodo principal. Si se omite este miembro de `ReplicaConfiguration`, ElastiCache para Redis OSS elegirá la zona de disponibilidad para cada una de las réplicas nuevas.

> **Important**
>
> Debe incluir únicamente uno de los parámetros `NewReplicaCount`, `ReplicasToRemove` o `ReplicaConfiguration`.

Example

En el siguiente ejemplo, se utiliza `NewReplicaCount` para reducir el número de réplicas del grupo de reproducción `sample-repl-group` a una. Al finalizar el ejemplo, existirá una réplica en cada grupo de nodos. Este número se aplica tanto si este es un grupo de Valkey o Redis OSS (modo de clúster deshabilitado) con un único grupo de nodos, como si es un grupo de Valkey o Redis OSS (modo de clúster habilitado) con varios grupos de nodos.

```
https://elasticache.us-west-2.amazonaws.com/
```

```
      ?Action=DecreaseReplicaCount
      &ApplyImmediately=True
      &NewReplicaCount=1
      &ReplicationGroupId=sample-repl-group
      &Version=2015-02-02
      &SignatureVersion=4
      &SignatureMethod=HmacSHA256
      &Timestamp=20150202T192317Z
      &X-Amz-Credential=<credential>
```

En el siguiente ejemplo, se reduce el número de réplicas del grupo de reproducción `sample-repl-group` eliminando dos réplicas especificadas (`0001` y `0003`) del grupo de nodos.

```
https://elasticache.us-west-2.amazonaws.com/
      ?Action=DecreaseReplicaCount
      &ApplyImmediately=True
      &ReplicasToRemove.ReplicaToRemove.1=0001
      &ReplicasToRemove.ReplicaToRemove.2=0003
      &ReplicationGroupId=sample-repl-group
      &Version=2015-02-02
      &SignatureVersion=4
      &SignatureMethod=HmacSHA256
      &Timestamp=20150202T192317Z
      &X-Amz-Credential=<credential>
```

En el siguiente ejemplo, se utiliza `ReplicaConfiguration` para reducir el número de réplicas del grupo de reproducción `sample-repl-group` al valor indicado para los dos grupos de nodos especificados. Dado que hay varios grupos de nodos, este es un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado). Al especificar el valor opcional `PreferredAvailabilityZones`, el número de zonas de disponibilidad indicadas debe ser igual al valor de `NewReplicaCount` más 1. Este enfoque incluye el nodo principal del grupo identificado por `NodeGroupId`.

```
https://elasticache.us-west-2.amazonaws.com/
      ?Action=DecreaseReplicaCount
      &ApplyImmediately=True
      &ReplicaConfiguration.ConfigureShard.1.NodeGroupId=0001
      &ReplicaConfiguration.ConfigureShard.1.NewReplicaCount=1

 &ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.1=
east-1a
```

```
 &ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.2=
east-1c
      &ReplicaConfiguration.ConfigureShard.2.NodeGroupId=0003
      &ReplicaConfiguration.ConfigureShard.2.NewReplicaCount=2

 &ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.1=
east-1a

 &ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.2=
east-1b

 &ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.4=
east-1c
      &ReplicationGroupId=sample-repl-group
      &Version=2015-02-02
      &SignatureVersion=4
      &SignatureMethod=HmacSHA256
      &Timestamp=20150202T192317Z
      &X-Amz-Credential=<credential>
```

Para obtener más información sobre cómo reducir el número de réplicas mediante la API, consulte DecreaseReplicaCount en la Referencia de la API de Amazon ElastiCache.

Adición de una réplica de lectura para Valkey o Redis OSS (modo de clúster deshabilitado)

La información del siguiente tema solo se aplica a los grupos de replicación de Valkey o Redis OSS (modo de clúster deshabilitado).

A medida que aumentan las operaciones de lectura, es recomendable que las distribuya entre más nodos para reducir así la presión de lectura en estos. En este tema, aprenderá a añadir una réplica de lectura a un clúster de Valkey o Redis OSS (modo de clúster deshabilitado).

Un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) puede tener un máximo de cinco réplicas de lectura. Si intenta agregar una réplica de lectura a un grupo de reproducción que ya tenga cinco réplicas de lectura, la operación produce un error.

Para obtener más información sobre la adición de réplicas a un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado), consulte los siguientes temas:

- Escalado de clústeres de Valkey o Redis OSS (modo de clúster habilitado)
- Aumento del número de réplicas de una partición

Puede añadir una réplica de lectura a un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) mediante la consola de ElastiCache, la AWS CLI o la API de ElastiCache.

Temas relacionados

- Adición de nodos a un clúster de ElastiCache
- Adición de una réplica de lectura a un grupo de reproducción (AWS CLI)
- Adición de una réplica de lectura a un grupo de reproducción mediante la API

Adición de una réplica de lectura a un grupo de reproducción (AWS CLI)

Para añadir una réplica de lectura a un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado), utilice el comando `create-cache-cluster` de la AWS CLI, con el parámetro `--replication-group-id` para especificar a qué grupo de replicación añadir el clúster (nodo).

El siguiente ejemplo crea el clúster `my-read replica` y lo añade al grupo de replicación `my-replication-group`. Los tipos de nodos, grupos de parámetros, grupos de seguridad, periodos de mantenimiento y otros ajustes de la réplica de lectura serán los mismos que los de los demás nodos de `my-replication-group`.

Para Linux, macOS o Unix:

```
aws elasticache create-cache-cluster \
      --cache-cluster-id my-read-replica \
      --replication-group-id my-replication-group
```

Para Windows:

```
aws elasticache create-cache-cluster ^
      --cache-cluster-id my-read-replica ^
      --replication-group-id my-replication-group
```

Para obtener más información sobre cómo agregar una réplica de lectura mediante la CLI, consulte create-cache-cluster en la Referencia de línea de comandos de Amazon ElastiCache.

Adición de una réplica de lectura a un grupo de reproducción mediante la API

Para añadir una réplica de lectura a un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado), utilice la operación `CreateCacheCluster` de ElastiCache, con el parámetro `ReplicationGroupId` para especificar a qué grupo de replicación añadir el clúster (nodo).

El siguiente ejemplo crea el clúster `myReadReplica` y lo añade al grupo de replicación `myReplicationGroup`. Los tipos de nodos, grupos de parámetros, grupos de seguridad, periodos de mantenimiento y otros ajustes de la réplica de lectura serán los mismos que los de los demás nodos de `myReplicationGroup`.

```
https://elasticache.us-west-2.amazonaws.com/
      ?Action=CreateCacheCluster
      &CacheClusterId=myReadReplica
      &ReplicationGroupId=myReplicationGroup
      &Version=2015-02-02
      &SignatureVersion=4
      &SignatureMethod=HmacSHA256
      &Timestamp=20150202T192317Z
      &X-Amz-Credential=<credential>
```

Para obtener más información sobre cómo agregar una réplica de lectura mediante la API, consulte CreateCacheCluster en la Referencia de la API de Amazon ElastiCache.

Eliminación de una réplica de lectura para Valkey o Redis OSS (modo de clúster deshabilitado)

La información del siguiente tema solo se aplica a los grupos de replicación de Valkey o Redis OSS (modo de clúster deshabilitado).

A medida que el tráfico del grupo de replicación de Valkey o Redis OSS cambie, puede que desee añadir o eliminar réplicas de lectura. Eliminar un nodo de un grupo de replicación de es igual que eliminar un clúster, aunque existen algunas restricciones:

- No puede eliminar el nodo principal de un grupo de reproducción. Si desea eliminar el nodo principal, haga lo siguiente:

  1. Promocione una réplica de lectura a principal. Para obtener más información acerca de cómo promocionar una réplica de lectura a principal, consulte Promoción de una réplica de lectura a principal para grupos de replicación de Valkey o Redis OSS (modo de clúster deshabilitado).
  2. Elimine el principal anterior. Consulte el siguiente punto para conocer una restricción de este método.
- Si se ha habilitado Multi-AZ en un grupo de reproducción, no se puede eliminar la última réplica de lectura del grupo de reproducción. En este caso, haga lo siguiente:

  1. Modifique el grupo de reproducción deshabilitando las Multi-AZ. Para obtener más información, consulte Modificación de un grupo de reproducción.

2. Elimine la réplica de lectura.

Puede eliminar una réplica de lectura de un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) mediante la consola de ElastiCache, la AWS CLI para ElastiCache o la API de ElastiCache.

Para obtener más información sobre cómo eliminar un clúster de un grupo de replicación de Valkey o Redis OSS, consulte los temas siguientes:

- Uso de Consola de administración de AWS
- Uso de AWS CLI para eliminar un clúster de ElastiCache
- Uso de la API de ElastiCache
- Escalado de clústeres de Valkey o Redis OSS (modo de clúster habilitado)
- Reducción del número de réplicas de una partición

Promoción de una réplica de lectura a principal para grupos de replicación de Valkey o Redis OSS (modo de clúster deshabilitado)

La información del siguiente tema solo se aplica solo a los grupos de replicación de Valkey o Redis OSS (modo de clúster deshabilitado).

Puede promover una réplica de lectura de Valkey o Redis OSS (modo de clúster deshabilitado) a un nodo principal mediante la Consola de administración de AWS, la AWS CLI o la API de ElastiCache. No se puede promocionar una réplica de lectura a principal mientras estén habilitadas múltiples zonas de disponibilidad con conmutación por error automática en el grupo de replicación de . Para promocionar una réplica de Valkey o Redis OSS (modo de clúster deshabilitado) a nodo principal en un grupo de replicación con multi-AZ habilitado, realice lo siguiente:

1. Modifique el grupo de reproducción para deshabilitar las Multi-AZ (esto no requiere que todos los clústeres estén en la misma zona de disponibilidad). Para obtener más información, consulte Modificación de un grupo de reproducción.
2. Promocione la réplica de lectura a principal.
3. Modifique el grupo de reproducción para volver a habilitar Multi-AZ.

Multi-AZ no está disponible en grupos de replicación que ejecutan Redis OSS 2.6.13 o versiones anteriores.

Uso de Consola de administración de AWS

El siguiente procedimiento utiliza la consola para promocionar un nodo de réplica a principal.

Para promocionar una réplica de lectura a nodo primario (consola)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. Si la réplica que desea promocionar es miembro de un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) con multi-AZ habilitado, debe modificar dicho grupo para deshabilitar multi-AZ antes de continuar. Para obtener más información, consulte Modificación de un grupo de reproducción.
3. Elija Valkey o Redis OSS y, a continuación, en la lista de clústeres, elija el grupo de replicación que desea modificar. Este grupo de reproducción debe ejecutar el motor "Redis", no el motor "Redis en clúster", y debe tener al menos dos nodos.

4. En la lista de nodos, elija el nodo de réplica que desea promocionar a principal y, a continuación, en Actions (Acciones), elija Promote (Promocionar).
5. En el cuadro de diálogo Promote Read Replica (Promocionar réplica de lectura), haga lo siguiente:

   a. En Apply Immediately (Aplicar inmediatamente), elija Yes (Sí) para promocionar la réplica de lectura inmediatamente, o No para promocionarla en el próximo periodo de mantenimiento del clúster.

   b. Elija Promote para promocionar la réplica de lectura o Cancel para cancelar la operación.
6. Si el clúster tenía Multi-AZ habilitadas antes de comenzar el proceso de promoción, espere hasta que el estado del grupo de reproducción sea available (disponible) y, a continuación, modifique el clúster para volver a habilitar las Multi-AZ. Para obtener más información, consulte Modificación de un grupo de reproducción.

Mediante AWS CLI

No puede promocionar una réplica de lectura a principal si el grupo de reproducción tiene Multi-AZ habilitadas. En algunos casos, la réplica que se desea promocionar puede ser miembro de un grupo de reproducción con Multi-AZ habilitadas. En estos casos, debe modificar el grupo de reproducción para deshabilitar la característica Multi-AZ antes de continuar. Esto no requiere que todos los clústeres estén en la misma zona de disponibilidad. Para obtener más información acerca de cómo modificar un grupo de reproducción, consulte Modificación de un grupo de reproducción.

El siguiente comando de la AWS CLI modifica el grupo de reproducción `sample-repl-group`, lo que hace que la réplica `my-replica-1` sea la principal en el grupo de reproducción.

Para Linux, macOS o Unix:

```
aws elasticache modify-replication-group \
   --replication-group-id sample-repl-group \
   --primary-cluster-id my-replica-1
```

Para Windows:

```
aws elasticache modify-replication-group ^
   --replication-group-id sample-repl-group ^
   --primary-cluster-id my-replica-1
```

Para obtener más información sobre la modificación de un grupo de reproducción, consulte modify-replication-group en la Referencia de línea de comandos de Amazon ElastiCache.

Uso de la API de ElastiCache

No puede promocionar una réplica de lectura a principal si el grupo de reproducción tiene Multi-AZ habilitadas. En algunos casos, la réplica que se desea promocionar puede ser miembro de un grupo de reproducción con Multi-AZ habilitadas. En estos casos, debe modificar el grupo de reproducción para deshabilitar la característica Multi-AZ antes de continuar. Esto no requiere que todos los clústeres estén en la misma zona de disponibilidad. Para obtener más información acerca de cómo modificar un grupo de reproducción, consulte Modificación de un grupo de reproducción.

La siguiente acción de la API de ElastiCache modifica el grupo de reproducción `myReplGroup`, lo que hace que la réplica de lectura `myReplica-1` sea el primario en el grupo de reproducción.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=ModifyReplicationGroup
   &ReplicationGroupId=myReplGroup
   &PrimaryClusterId=myReplica-1
   &Version=2014-12-01
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20141201T220302Z
   &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
   &X-Amz-Date=20141201T220302Z
   &X-Amz-SignedHeaders=Host
   &X-Amz-Expires=20141201T220302Z
   &X-Amz-Credential=<credential>
   &X-Amz-Signature=<signature>
```

Para obtener más información sobre la modificación de un grupo de reproducción, consulte ModifyReplicationGroup en la Referencia de la API de Amazon ElastiCache.

Administración del mantenimiento de clústeres de ElastiCache

Cada clúster tiene un periodo de mantenimiento semanal durante el que se aplican los cambios del sistema. Con Valkey y Redis OSS, los grupos de replicación tienen el mismo período de mantenimiento semanal. Si no especifica un periodo de mantenimiento preferido al crear o modificar un clúster o grupo de reproducción, ElastiCache asigna un periodo de mantenimiento de 60 minutos en el periodo de mantenimiento de su región en un día de la semana elegido al azar.

El periodo de mantenimiento de 60 minutos se elige al azar de un bloque de 8 horas por cada región. En la siguiente tabla, se muestran los bloques de tiempo de cada región desde los que se asignan los periodos predeterminados de mantenimiento. Puede elegir un periodo de mantenimiento preferido fuera del bloque del periodo de mantenimiento de la región.

Código de región	Nombre de la región	Periodo de mantenimiento de la región
ap-northeast-1	Región Asia-Pacífico (Tokio)	13:00-21:00 UTC
ap-northeast-2	Región Asia-Pacífico (Seúl)	12:00-20:00 UTC
ap-northeast-3	Región Asia-Pacífico (Osaka)	12:00-20:00 UTC
ap-southeast-3	Región Asia-Pacífico (Yakarta)	14:00-22:00 UTC
ap-south-1	Región Asia-Pacífico (Mumbai)	17:30-01:30 UTC
ap-southeast-1	Región Asia-Pacífico (Singapur)	14:00-22:00 UTC
cn-north-1	Región China (Pekín)	14:00-22:00 UTC
cn-northwest-1	Región China (Ningxia)	14:00-22:00 UTC
ap-east-1	Región de Asia-Pacífico (Hong Kong)	13:00-21:00 UTC
ap-southeast-2	Región Asia-Pacífico (Sídney)	12:00-20:00 UTC
eu-west-3	Región EU (París)	23:59-07:29 UTC
af-south-1	Región África (Ciudad del Cabo)	13:00-21:00 UTC
eu-central-1	Región de Europa (Fráncfort)	23:00-07:00 UTC
eu-west-1	Región de Europa (Irlanda)	22:00-06:00 UTC
eu-west-2	Región de Europa (Londres)	23:00-07:00 UTC
me-south-1	Región Medio Oriente (Baréin)	13:00-21:00 UTC

Código de región	Nombre de la región	Periodo de mantenimiento de la región
me-central-1	Región de Oriente Medio (EAU)	13:00-21:00 UTC
eu-south-1	Región Europa (Milán)	21:00 — 05:00 UTC
sa-east-1	Región de América del Sur (São Paulo)	01:00-09:00 UTC
us-east-1	Región Este de EE. UU. (Norte de Virginia)	03:00-11:00 UTC
us-east-2	Región del este de EE. UU. (Ohio)	04:00 a 12:00 h UTC
us-gov-west-1	AWS GovCloud (US) región	06:00-14:00 UTC
us-west-1	Región Oeste de EE. UU. (Norte de California)	06:00-14:00 UTC
us-west-2	Región del Oeste de EE. UU (Oregón)	06:00-14:00 UTC

Cambio del periodo de mantenimiento del clúster o grupo de replicación

El periodo de mantenimiento debe corresponder al momento de mínimo uso y, por tanto, podría ser preciso modificarlo cada cierto tiempo. Puede modificar el clúster o grupo de reproducción de modo que especifique un intervalo de tiempo de hasta 24 horas durante las cuales deban llevarse a cabo todas las actividades de mantenimiento que solicite. Las modificaciones de clúster pendientes o aplazadas que ha solicitado tendrán lugar en este periodo.

> **Note**
>
> Si desea aplicar modificaciones del tipo de nodo o actualizaciones de motor de forma inmediata con la Consola de administración de AWS seleccione el cuadro Aplicar ahora. De lo contrario, estas modificaciones se aplicarán durante el siguiente periodo de mantenimiento programado. Para utilizar la API, consulte modify-replication-group o modify-cache-cluster.

Más información

Para obtener más información sobre el periodo de mantenimiento y de la sustitución de nodos, consulte lo siguiente:

- Mantenimiento de ElastiCache: preguntas frecuentes sobre el mantenimiento y la sustitución de nodos
- Reemplazo de nodos (Memcached): administración del reemplazo de nodos para Memcached
- Modificación de un clúster de ElastiCache: cambio del periodo de mantenimiento del clúster
- Reemplazo de nodos (Valkey y Redis OSS): administración de la sustitución de nodos
- Modificación de un grupo de reproducción: cambio del periodo de mantenimiento de un grupo de reproducción

Configuración de los parámetros de motor mediante grupos de parámetros de ElastiCache

Amazon ElastiCache utiliza parámetros para controlar las propiedades de tiempo de ejecución de sus nodos y clústeres. Por lo general, las versiones de motor más reciente incluyen parámetros adicionales para ofrecer compatibilidad con la funcionalidad más reciente. Para ver tablas de parámetros de Memcached, consulte Parámetros específicos de Memcached. Para ver tablas de parámetros de Valkey y Redis OSS, consulte Parámetros de Valkey y Redis OSS.

Como cabe esperar, determinados valores de parámetros, como `maxmemory`, dependen del tipo de nodo y de motor. Para ver una tabla de valores de estos parámetros de Memcached por tipo de nodo, consulte Parámetros específicos de tipo de nodo de Memcached. Para ver una tabla de valores de estos parámetros de Valkey y Redis OSS por tipo de nodo, consulte Parámetros específicos de tipos de nodos de Redis OSS.

> **Note**
>
> Para obtener una lista de parámetros específicos de Memcached, consulte Parámetros específicos de Memcached.

Temas

- Administración de parámetros en ElastiCache

- Niveles de grupo de parámetros de caché en ElastiCache
- Creación de un grupo de parámetros de ElastiCache
- Enumeración de grupos de parámetros de ElastiCache por nombre
- Enumeración de valores de un grupo de parámetros de ElastiCache
- Modificación de un grupo de parámetros de ElastiCache
- Eliminación de un grupo de parámetros de ElastiCache
- Parámetros específicos del motor

Administración de parámetros en ElastiCache

Los parámetros de ElastiCache se agrupan en grupos de parámetros con nombre para facilitar su administración. Un grupo de parámetros representa una combinación de valores específicos de parámetros que se pasan al software del motor durante el startup. Estos valores determinan cómo se comportan los procesos del motor en cada nodo durante el tiempo de ejecución. Los valores de parámetros de un grupo de parámetros determinado se aplican a todos los nodos asociados al grupo, independientemente del clúster al que pertenezcan.

Para refinar el rendimiento del clúster, puede modificar los valores de algunos parámetros o cambiar el grupo de parámetros del clúster.

- No puede modificar ni eliminar los grupos de parámetros predeterminados. Si necesita valores de parámetros personalizados, debe crear un grupo de parámetros personalizado.
- Para Memcached, la familia del grupo parámetros y el clúster que va a asignar deben ser compatibles. Por ejemplo, si el clúster está ejecutando Memcached versión 1.4.8, solo se pueden utilizar los grupos de parámetros, predeterminados o personalizados, de la familia Memcached 1.4.

  Para Redis OSS, la familia del grupo parámetros y el clúster que va a asignar deben ser compatibles. Por ejemplo, si el clúster está ejecutando Redis OSS versión 3.2.10, solo se pueden utilizar los grupos de parámetros, ya sean predeterminados o personalizados, de la familia de Redis OSS 3.2.
- Si cambia el grupo de parámetros de un clúster, los valores de los parámetros modificables condicionalmente deben ser los mismos tanto en los grupos de parámetros actuales como en los nuevos.
- Para Memcached, cuando cambia los parámetros de un clúster, el cambio se aplica al clúster inmediatamente. Esto es cierto tanto si se modifica el propio grupo de parámetros del clúster como si se modifica el valor de un parámetro del grupo. Para determinar cuándo se aplica un cambio de parámetros determinado, consulte la columna Aplicación de los cambios en las tablas de Parámetros específicos de Memcached. Para obtener información sobre cómo reiniciar los nodos de un clúster, consulte Rebooting clusters.
- Para Redis OSS, cuando se modifican los parámetros de un clúster, el cambio se aplica al clúster de inmediato o, con las excepciones indicadas a continuación, después de que se reinicien los nodos del clúster. Esto es cierto tanto si se modifica el propio grupo de parámetros del clúster como si se modifica el valor de un parámetro del grupo. Para determinar cuándo se aplica un cambio de parámetros determinado, consulte la columna Aplicación de los cambios en las tablas de Parámetros de Valkey y Redis OSS.

Para obtener más información sobre cómo reiniciar nodos de Valkey o Redis OSS, consulte Reinicio de nodos.

> **Cambios de parámetros de Valkey o Redis OSS (modo de clúster habilitado)**
>
> Si realiza cambios en los siguientes parámetros en un clúster de Valkey o Redis OSS (modo de clúster habilitado), realice los siguientes pasos.
>
> - activerehashing
> - bases de datos
>
> 1. Cree una copia de seguridad manual de su clúster. Consulte Copias de seguridad manuales.
> 2. Eliminar el clúster. Consulte Deleting clusters.
> 3. almacenar el clúster con la copia de seguridad y el grupo de parámetros modificados para propagar los datos en el nuevo clúster. Consulte Restauración desde una copia de seguridad a una nueva caché.
>
> Los cambios en otros parámetros no requieren esta acción.

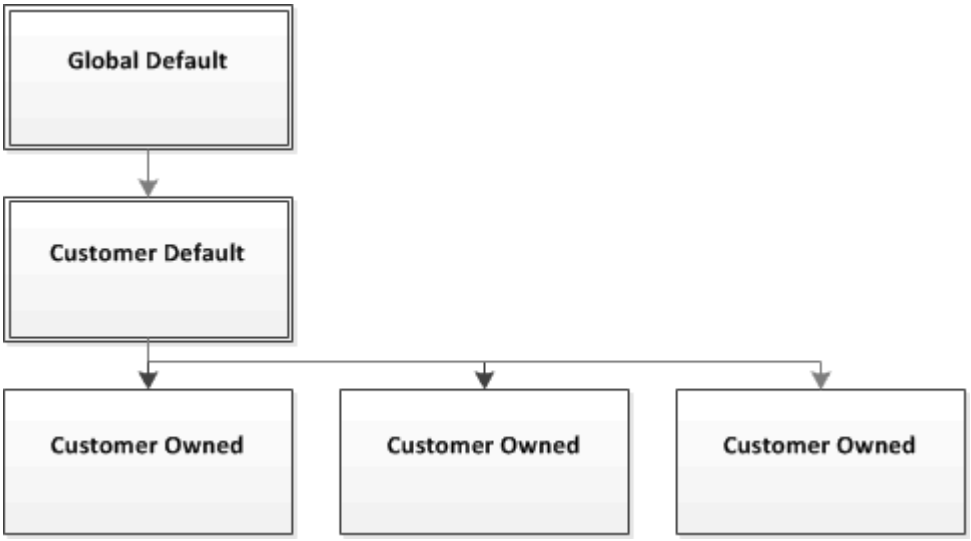
- Puede asociar grupos de parámetros con almacenes de datos globales de Valkey o Redis OSS. Los almacenes de datos globales son una recopilación de uno o varios clústeres que abarcan regiones de AWS. En este caso, el grupo de parámetros lo comparten todos los clústeres que conforman el almacén de datos global. Cualquier modificación del grupo de parámetros del clúster principal se replica en todos los clústeres restantes del almacén de datos global. Para obtener más información, consulte Replicación en regiones de AWS mediante los almacenes de datos globales.

  Puede comprobar si un grupo de parámetros forma parte de un almacén de datos global buscando en estas ubicaciones:

  - En la consola de ElastiCache en la página de Parameter Groups (Grupos de parámetros), el atributo sí/no Global.
  - La propiedad sí/no `IsGlobal` de la operación de la API CacheParameterGroup

Niveles de grupo de parámetros de caché en ElastiCache

Amazon ElastiCache tiene tres niveles de grupos de parámetros de caché como se muestra a continuación.



Niveles de grupo de parámetros de Amazon ElastiCache

Predeterminado global

Este es el grupo de parámetros raíz de nivel superior para todos los clientes de Amazon ElastiCache de la región.

Características del grupo de parámetros de caché Predeterminado global:

- Se encuentra reservado para ElastiCache y no se encuentra disponible para el cliente.

Predeterminado del cliente

Se trata de una copia del grupo de parámetros de cache Predeterminado global que se crea para el uso del cliente.

Características del grupo de parámetros de caché Predeterminado del cliente:

- Lo crea ElastiCache y es de su propiedad.
- Está disponible para el cliente para el uso como grupo de parámetros de caché para cualquier clúster que ejecute una versión de motor compatible con este grupo de parámetros de caché.
- No admite modificación del cliente.

Propiedad del cliente

Se trata de una copia del grupo de parámetros de caché predeterminado del cliente. El grupo de parámetros de caché Propiedad del cliente se crea cuando el cliente crea un grupo de parámetros de caché.

Características del grupo de parámetros de caché Propiedad del cliente:

- Lo crea el cliente y es de su propiedad.
- Puede asignarse a cualquiera de los clústeres compatibles del cliente.
- Lo puede modificar el cliente para crear un nuevo grupo de parámetros de caché personalizado.

  No todos los valores de parámetros se pueden modificar. Para obtener más información, sobre los valores de Memcached, consulte Parámetros específicos de Memcached. Para obtener más información sobre los valores de Valkey y Redis OSS, consulte Parámetros de Valkey y Redis OSS.

Creación de un grupo de parámetros de ElastiCache

Debe crear un nuevo grupo de parámetros si existe uno o varios parámetros que desee cambiar con respecto a los valores predeterminados. Puede crear un grupo de parámetros con la consola de ElastiCache, la AWS CLI o la API de ElastiCache.

Creación de un grupo de parámetros de ElastiCache (consola)

En el siguiente procedimiento se muestra cómo crear un grupo de parámetros mediante la consola de ElastiCache.

Para crear un grupo de parámetros con la consola de ElastiCache

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. Para ver una lista de todos los grupos de parámetros disponibles, en el panel de navegación izquierdo, elija Parameter Groups.
3. Para crear un grupo de parámetros, elija Create Parameter Group.

   Aparece la ventana de Create Parameter Group (Crear grupo de parámetros).
4. En la lista Family, elija la familia del grupo de parámetros que será la plantilla de su grupo de parámetros.

La familia de grupos de parámetros, como memcached1.4 o redis3.2, define los parámetros reales del grupo de parámetros, así como sus valores iniciales. La familia del grupo de parámetros debe coincidir con la versión y el motor del clúster.

5. En el cuadro Name, escriba un nombre único para este grupo de parámetros.

   Al crear un clúster o modificar un grupo de parámetros de clúster, podrá elegir el grupo de parámetros por su nombre. Por lo tanto, se recomienda que el nombre sea informativo y que identifique de algún modo la familia del grupo de parámetros.

   Las restricciones de nomenclatura de los grupos de parámetros son las siguientes:

   - Deben comenzar por una letra ASCII.
   - Solo puede contener letras ASCII, dígitos y guiones.
   - Debe tener de 1 a 255 caracteres.
   - No pueden contener dos guiones consecutivos.
   - No pueden terminar con un guion.
6. En el cuadro Description, escriba una descripción para el grupo de parámetros.
7. Para crear el grupo de parámetros, elija Create.

   Para finalizar el proceso sin crear el grupo de parámetros, seleccione Cancel.
8. Cuando se cree el grupo de parámetros, tendrá los valores predeterminados de la familia. Para cambiar los valores predeterminados, debe modificar el grupo de parámetros. Para obtener más información, consulte Modificación de un grupo de parámetros de ElastiCache.

Creación de un grupo de parámetros de ElastiCache (AWS CLI)

Para crear un grupo de parámetros con la AWS CLI, use el comando `create-cache-parameter-group` con los parámetros que se indican a continuación.

- `--cache-parameter-group-name`: el nombre del grupo de parámetros.

  Las restricciones de nomenclatura de los grupos de parámetros son las siguientes:
  - Deben comenzar por una letra ASCII.
  - Solo puede contener letras ASCII, dígitos y guiones.
  - Debe tener de 1 a 255 caracteres.
  - No pueden contener dos guiones consecutivos.

  - No pueden terminar con un guion.
- `--cache-parameter-group-family`: la familia de versión y motor del grupo de parámetros.
- `--description`: una descripción del usuario para el grupo de parámetros.

Example

En el ejemplo siguiente, se crea un grupo de parámetros denominado myMem14 que usa la familia memcached1.4 como plantilla.

Para Linux, macOS o Unix:

```
aws elasticache create-cache-parameter-group \
    --cache-parameter-group-name myMem14  \
    --cache-parameter-group-family memcached1.4 \
    --description "My first parameter group"
```

Para Windows:

```
aws elasticache create-cache-parameter-group ^
    --cache-parameter-group-name myMem14  ^
    --cache-parameter-group-family memcached1.4 ^
    --description "My first parameter group"
```

La salida de este comando será similar a lo que se muestra a continuación.

```
{
    "CacheParameterGroup": {
        "CacheParameterGroupName": "myMem14",
        "CacheParameterGroupFamily": "memcached1.4",
        "Description": "My first  parameter group"
    }
}
```

Example

En el ejemplo siguiente, se crea un grupo de parámetros denominado myRed28 que usa la familia redis2.8 como plantilla.

Para Linux, macOS o Unix:

```
aws elasticache create-cache-parameter-group \
    --cache-parameter-group-name myRed28  \
    --cache-parameter-group-family redis2.8 \
    --description "My first parameter group"
```

Para Windows:

```
aws elasticache create-cache-parameter-group ^
    --cache-parameter-group-name myRed28  ^
    --cache-parameter-group-family redis2.8 ^
    --description "My first parameter group"
```

La salida de este comando será similar a lo que se muestra a continuación.

```
{
    "CacheParameterGroup": {
        "CacheParameterGroupName": "myRed28",
        "CacheParameterGroupFamily": "redis2.8",
        "Description": "My first parameter group"
    }
}
```

Cuando se cree el grupo de parámetros, tendrá los valores predeterminados de la familia. Para cambiar los valores predeterminados, debe modificar el grupo de parámetros. Para obtener más información, consulte Modificación de un grupo de parámetros de ElastiCache.

Para obtener más información, consulte `create-cache-parameter-group`.

Creación de un grupo de parámetros de ElastiCache (API de ElastiCache)

Para crear un grupo de parámetros con la API de ElastiCache, utilice la acción `CreateCacheParameterGroup` con los parámetros que se indican a continuación.

- `ParameterGroupName`: el nombre del grupo de parámetros.

  Las restricciones de nomenclatura de los grupos de parámetros son las siguientes:

  - Deben comenzar por una letra ASCII.
  - Solo puede contener letras ASCII, dígitos y guiones.
  - Debe tener de 1 a 255 caracteres.

  - No pueden contener dos guiones consecutivos.
  - No pueden terminar con un guion.
- `CacheParameterGroupFamily`: la familia de versión y motor del grupo de parámetros. Por ejemplo, `memcached1.4`.
- `CacheParameterGroupFamily`: la familia de versión y motor del grupo de parámetros. Por ejemplo, `redis2.8`.
- `Description`: una descripción del usuario para el grupo de parámetros.

Example

En el ejemplo siguiente, se crea un grupo de parámetros denominado myMem14 que usa la familia memcached1.4 como plantilla.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=CreateCacheParameterGroup
   &CacheParameterGroupFamily=memcached1.4
   &CacheParameterGroupName=myMem14
   &Description=My%20first%20parameter%20group
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &Version=2015-02-02
   &X-Amz-Credential=<credential>
```

La respuesta a esta acción será similar a lo que se muestra a continuación.

```
<CreateCacheParameterGroupResponse xmlns="http://elasticache.amazonaws.com/
doc/2013-06-15/">
  <CreateCacheParameterGroupResult>
    <CacheParameterGroup>
      <CacheParameterGroupName>myMem14</CacheParameterGroupName>
      <CacheParameterGroupFamily>memcached1.4</CacheParameterGroupFamily>
      <Description>My first  parameter group</Description>
    </CacheParameterGroup>
  </CreateCacheParameterGroupResult>
  <ResponseMetadata>
    <RequestId>d8465952-af48-11e0-8d36-859edca6f4b8</RequestId>
  </ResponseMetadata>
</CreateCacheParameterGroupResponse>
```

Example

En el ejemplo siguiente, se crea un grupo de parámetros denominado myRed28 que usa la familia redis2.8 como plantilla.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=CreateCacheParameterGroup
   &CacheParameterGroupFamily=redis2.8
   &CacheParameterGroupName=myRed28
   &Description=My%20first%20parameter%20group
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &Version=2015-02-02
   &X-Amz-Credential=<credential>
```

La respuesta a esta acción será similar a lo que se muestra a continuación.

```
<CreateCacheParameterGroupResponse xmlns="http://elasticache.amazonaws.com/
doc/2013-06-15/">
  <CreateCacheParameterGroupResult>
    <CacheParameterGroup>
      <CacheParameterGroupName>myRed28</CacheParameterGroupName>
      <CacheParameterGroupFamily>redis2.8</CacheParameterGroupFamily>
      <Description>My first parameter group</Description>
    </CacheParameterGroup>
  </CreateCacheParameterGroupResult>
  <ResponseMetadata>
    <RequestId>d8465952-af48-11e0-8d36-859edca6f4b8</RequestId>
  </ResponseMetadata>
</CreateCacheParameterGroupResponse>
```

Cuando se cree el grupo de parámetros, tendrá los valores predeterminados de la familia. Para cambiar los valores predeterminados, debe modificar el grupo de parámetros. Para obtener más información, consulte Modificación de un grupo de parámetros de ElastiCache.

Para obtener más información, consulte `CreateCacheParameterGroup`.

Enumeración de grupos de parámetros de ElastiCache por nombre

Puede enumerar los grupos de parámetros con la consola de ElastiCache, la AWS CLI o la API de ElastiCache.

Enumeración de grupos de parámetros por nombre (consola)

En el siguiente procedimiento se muestra cómo ver una lista de grupos de parámetros mediante la consola de ElastiCache.

Para obtener una lista de grupos de parámetros mediante la consola de ElastiCache

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. Para ver una lista de todos los grupos de parámetros disponibles, en el panel de navegación izquierdo, elija Parameter Groups.

Enumeración de grupos de parámetros de ElastiCache por nombre (AWS CLI)

Para generar una lista de grupos de parámetros mediante la AWS CLI, use el comando `describe-cache-parameter-groups`. Si proporciona un nombre de grupo de parámetros, solo se mostrará el grupo de parámetros de dicho nombre. Si no proporciona ningún nombre de grupo de parámetros, se mostrarán hasta `--max-records` grupos de parámetros. En cualquier caso, se mostrarán el nombre, la familia y la descripción del grupo de parámetros.

Example

El siguiente código de ejemplo muestra el grupo de parámetros myMem14.

Para Linux, macOS o Unix:

```
aws elasticache describe-cache-parameter-groups \
    --cache-parameter-group-name myMem14
```

Para Windows:

```
aws elasticache describe-cache-parameter-groups ^
    --cache-parameter-group-name myMem14
```

La salida de este comando tendrá un aspecto similar al siguiente y mostrará el nombre, la familia y la descripción del grupo de parámetros.

```
{
    "CacheParameterGroups": [
     {
         "CacheParameterGroupName": "myMem14",
         "CacheParameterGroupFamily": "memcached1.4",
         "Description": "My first parameter group"
     }
    ]
}
```

Example

El siguiente código de ejemplo muestra el grupo de parámetros myRed28.

Para Linux, macOS o Unix:

```
aws elasticache describe-cache-parameter-groups \
    --cache-parameter-group-name myRed28
```

Para Windows:

```
aws elasticache describe-cache-parameter-groups ^
    --cache-parameter-group-name myRed28
```

La salida de este comando tendrá un aspecto similar al siguiente y mostrará el nombre, la familia y la descripción del grupo de parámetros.

```
{
    "CacheParameterGroups": [
     {
         "CacheParameterGroupName": "myRed28",
         "CacheParameterGroupFamily": "redis2.8",
         "Description": "My first parameter group"
     }
    ]
}
```

Example

El siguiente código de ejemplo indica el grupo de parámetros myRed56 para grupos de parámetros que se ejecutan en la versión 5.0.6 en adelante del motor de Redis OSS. Si el grupo de parámetros

forma parte de un Replicación en regiones de AWS mediante los almacenes de datos globales, el valor de propiedad `IsGlobal` devuelto en la salida será `Yes`.

Para Linux, macOS o Unix:

```
aws elasticache describe-cache-parameter-groups \
    --cache-parameter-group-name myRed56
```

Para Windows:

```
aws elasticache describe-cache-parameter-groups ^
    --cache-parameter-group-name myRed56
```

La salida de este comando tendrá un aspecto similar al siguiente y mostrará el nombre, la familia, isGlobal y la descripción del grupo de parámetros.

```
{
    "CacheParameterGroups": [
     {
         "CacheParameterGroupName": "myRed56",
         "CacheParameterGroupFamily": "redis5.0",
         "Description": "My first parameter group",
         "IsGlobal": "yes"
     }
    ]
}
```

Example

El siguiente código de ejemplo muestra hasta 10 grupos de parámetros.

```
aws elasticache describe-cache-parameter-groups --max-records 10
```

La salida JSON de este comando tendrá un aspecto similar al siguiente y mostrará el nombre, la familia, la descripción y, en el caso de redis5.6, si el grupo de parámetros forma parte de un almacén de datos global (isGlobal), para cada grupo de parámetros.

```
{
    "CacheParameterGroups": [
```

```
        {
            "CacheParameterGroupName": "custom-redis32",
            "CacheParameterGroupFamily": "redis3.2",
            "Description": "custom parameter group with reserved-memory > 0"
        },
        {
            "CacheParameterGroupName": "default.memcached1.4",
            "CacheParameterGroupFamily": "memcached1.4",
            "Description": "Default parameter group for memcached1.4"
        },
        {
            "CacheParameterGroupName": "default.redis2.6",
            "CacheParameterGroupFamily": "redis2.6",
            "Description": "Default parameter group for redis2.6"
        },
        {
            "CacheParameterGroupName": "default.redis2.8",
            "CacheParameterGroupFamily": "redis2.8",
            "Description": "Default parameter group for redis2.8"
        },
        {
            "CacheParameterGroupName": "default.redis3.2",
            "CacheParameterGroupFamily": "redis3.2",
            "Description": "Default parameter group for redis3.2"
        },
        {
            "CacheParameterGroupName": "default.redis3.2.cluster.on",
            "CacheParameterGroupFamily": "redis3.2",
            "Description": "Customized default parameter group for redis3.2 with
 cluster mode on"
        },
        {
            "CacheParameterGroupName": "default.redis5.6.cluster.on",
            "CacheParameterGroupFamily": "redis5.0",
            "Description": "Customized default parameter group for redis5.6 with
 cluster mode on",
            "isGlobal": "yes"
        },
    ]
}
```

Para obtener más información, consulte describe-cache-parameter-groups.

Enumeración de grupos de parámetros de ElastiCache por nombre (API de ElastiCache)

Para generar una lista de grupos de parámetros mediante la API de ElastiCache, utilice la acción `DescribeCacheParameterGroups`. Si proporciona un nombre de grupo de parámetros, solo se mostrará el grupo de parámetros de dicho nombre. Si no proporciona ningún nombre de grupo de parámetros, se mostrarán hasta `MaxRecords` grupos de parámetros. En cualquier caso, se mostrarán el nombre, la familia y la descripción del grupo de parámetros.

Example

El siguiente código de ejemplo muestra el grupo de parámetros myMem14.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DescribeCacheParameterGroups
   &CacheParameterGroupName=myMem14
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &Version=2015-02-02
   &X-Amz-Credential=<credential>
```

La respuesta de esta acción tendrá un aspecto similar al siguiente y mostrará el nombre, la familia y la descripción de cada grupo de parámetros.

```
<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/
doc/2013-06-15/">
  <DescribeCacheParameterGroupsResult>
    <CacheParameterGroups>
      <CacheParameterGroup>
        <CacheParameterGroupName>myMem14</CacheParameterGroupName>
        <CacheParameterGroupFamily>memcached1.4</CacheParameterGroupFamily>
        <Description>My custom Memcached 1.4 parameter group</Description>
      </CacheParameterGroup>
    </CacheParameterGroups>
  </DescribeCacheParameterGroupsResult>
  <ResponseMetadata>
    <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
  </ResponseMetadata>
</DescribeCacheParameterGroupsResponse>
```

Example

El siguiente código de ejemplo muestra hasta 10 grupos de parámetros.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DescribeCacheParameterGroups
   &MaxRecords=10
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &Version=2015-02-02
   &X-Amz-Credential=<credential>
```

La respuesta de esta acción tendrá un aspecto similar al siguiente y mostrará el nombre, la familia, la descripción y, en el caso de redis5.6, si el grupo de parámetros pertenece a un almacén de datos global (isGlobal), para cada grupo de parámetros.

```
<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/
doc/2013-06-15/">
  <DescribeCacheParameterGroupsResult>
    <CacheParameterGroups>
      <CacheParameterGroup>
        <CacheParameterGroupName>myRedis28</CacheParameterGroupName>
        <CacheParameterGroupFamily>redis2.8</CacheParameterGroupFamily>
        <Description>My custom Redis 2.8 parameter group</Description>
      </CacheParameterGroup>
      <CacheParameterGroup>
        <CacheParameterGroupName>myMem14</CacheParameterGroupName>
        <CacheParameterGroupFamily>memcached1.4</CacheParameterGroupFamily>
        <Description>My custom Memcached 1.4 parameter group</Description>
      </CacheParameterGroup>
       <CacheParameterGroup>
        <CacheParameterGroupName>myRedis56</CacheParameterGroupName>
        <CacheParameterGroupFamily>redis5.0</CacheParameterGroupFamily>
        <Description>My custom redis 5.6 parameter group</Description>
        <isGlobal>yes</isGlobal>
      </CacheParameterGroup>
    </CacheParameterGroups>
  </DescribeCacheParameterGroupsResult>
  <ResponseMetadata>
    <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
  </ResponseMetadata>
```

```
</DescribeCacheParameterGroupsResponse>
```

Example

El siguiente código de ejemplo muestra el grupo de parámetros myRed28.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DescribeCacheParameterGroups
   &CacheParameterGroupName=myRed28
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &Version=2015-02-02
   &X-Amz-Credential=<credential>
```

La respuesta de esta acción tendrá un aspecto similar al siguiente y mostrará el nombre, la familia y la descripción.

```
<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/
doc/2013-06-15/">
  <DescribeCacheParameterGroupsResult>
    <CacheParameterGroups>
      <CacheParameterGroup>
        <CacheParameterGroupName>myRed28</CacheParameterGroupName>
        <CacheParameterGroupFamily>redis2.8</CacheParameterGroupFamily>
        <Description>My custom Redis 2.8 parameter group</Description>
      </CacheParameterGroup>
    </CacheParameterGroups>
  </DescribeCacheParameterGroupsResult>
  <ResponseMetadata>
    <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
  </ResponseMetadata>
</DescribeCacheParameterGroupsResponse>
```

Example

El siguiente código de muestra indica el grupo de parámetros myRed56.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DescribeCacheParameterGroups
   &CacheParameterGroupName=myRed56
   &SignatureVersion=4
```

```
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &Version=2015-02-02
   &X-Amz-Credential=<credential>
```

La respuesta de esta acción tendrá un aspecto similar al siguiente y mostrará el nombre, la familia, la descripción y si el grupo de parámetros forma parte de un almacén de datos global (isGlobal).

```
<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/
doc/2013-06-15/">
  <DescribeCacheParameterGroupsResult>
    <CacheParameterGroups>
      <CacheParameterGroup>
        <CacheParameterGroupName>myRed56</CacheParameterGroupName>
        <CacheParameterGroupFamily>redis5.0</CacheParameterGroupFamily>
        <Description>My custom Redis 5.6 parameter group</Description>
        <isGlobal>yes</isGlobal>
      </CacheParameterGroup>
    </CacheParameterGroups>
  </DescribeCacheParameterGroupsResult>
  <ResponseMetadata>
    <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
  </ResponseMetadata>
</DescribeCacheParameterGroupsResponse>
```

Para obtener más información, consulte DescribeCacheParameterGroups.

Enumeración de valores de un grupo de parámetros de ElastiCache

Puede obtener una lista de los parámetros de un grupo de parámetros, junto con sus valores, mediante la consola de ElastiCache, la AWS CLI o la API de ElastiCache.

Enumeración de valores de un grupo de parámetros de ElastiCache (consola)

El siguiente procedimiento muestra cómo obtener una lista de los parámetros de un grupo de parámetros, junto con sus valores, mediante la consola de ElastiCache.

Para obtener una lista de los parámetros de un grupo de parámetros, junto con sus valores, mediante la consola de ElastiCache

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. Para ver una lista de todos los grupos de parámetros disponibles, en el panel de navegación izquierdo, elija Parameter Groups.
3. Elija el grupo de parámetros del que desea obtener una lista de los parámetros y sus valores activando la casilla situada a la izquierda del nombre del grupo de parámetros.

   Los parámetros y sus valores se mostrarán en la parte inferior de la pantalla. Debido al número de parámetros, puede que tenga que desplazarse hacia arriba y hacia abajo para encontrar el parámetro que le interesa.

Enumeración de valores de un grupo de parámetros (AWS CLI)

Para obtener una lista de los parámetros de un grupo de parámetros, junto con sus valores, mediante la AWS CLI, use el comando `describe-cache-parameters`.

Example

El siguiente código de ejemplo muestra todos los parámetros de Memcached, junto con sus valores, para el grupo de parámetros myMem14.

Para Linux, macOS o Unix:

```
aws elasticache describe-cache-parameters \
    --cache-parameter-group-name myMem14
```

Para Windows:

```
aws elasticache describe-cache-parameters ^
    --cache-parameter-group-name myMem14
```

Example

El siguiente código de ejemplo muestra todos los parámetros, junto con sus valores, del grupo de parámetros myRedis28.

Para Linux, macOS o Unix:

```
aws elasticache describe-cache-parameters \
    --cache-parameter-group-name myRedis28
```

Para Windows:

```
aws elasticache describe-cache-parameters ^
    --cache-parameter-group-name myRed28
```

Para obtener más información, consulte describe-cache-parameters.

Enumeración de valores de un grupo de parámetros (API de ElastiCache)

Para obtener una lista de los parámetros de un grupo de parámetros, junto con sus valores, mediante la API de ElastiCache, utilice la acción `DescribeCacheParameters`.

Example

El siguiente código de ejemplo muestra todos los parámetros de Memcached para el grupo de parámetros myMem14.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DescribeCacheParameters
   &CacheParameterGroupName=myMem14
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &Version=2015-02-02
   &X-Amz-Credential=<credential>
```

La respuesta a esta acción tendrá un aspecto similar al siguiente. Esta respuesta se ha truncado.

```
<DescribeCacheParametersResponse xmlns="http://elasticache.amazonaws.com/
doc/2013-06-15/">
  <DescribeCacheParametersResult>
    <CacheClusterClassSpecificParameters>
      <CacheNodeTypeSpecificParameter>
        <DataType>integer</DataType>
        <Source>system</Source>
        <IsModifiable>false</IsModifiable>
        <Description>The maximum configurable amount of memory to use to store items,
 in megabytes.</Description>
        <CacheNodeTypeSpecificValues>
          <CacheNodeTypeSpecificValue>
            <Value>1000</Value>
            <CacheClusterClass>cache.c1.medium</CacheClusterClass>
          </CacheNodeTypeSpecificValue>
          <CacheNodeTypeSpecificValue>
            <Value>6000</Value>
            <CacheClusterClass>cache.c1.xlarge</CacheClusterClass>
          </CacheNodeTypeSpecificValue>
          <CacheNodeTypeSpecificValue>
            <Value>7100</Value>
            <CacheClusterClass>cache.m1.large</CacheClusterClass>
          </CacheNodeTypeSpecificValue>
          <CacheNodeTypeSpecificValue>
            <Value>1300</Value>
            <CacheClusterClass>cache.m1.small</CacheClusterClass>
          </CacheNodeTypeSpecificValue>

...output omitted...

    </CacheClusterClassSpecificParameters>
  </DescribeCacheParametersResult>
  <ResponseMetadata>
    <RequestId>6d355589-af49-11e0-97f9-279771c4477e</RequestId>
  </ResponseMetadata>
</DescribeCacheParametersResponse>
```

Example

El siguiente código de ejemplo muestra todos los parámetros del grupo de parámetros myRed28.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DescribeCacheParameters
```

```
   &CacheParameterGroupName=myRed28
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &Version=2015-02-02
   &X-Amz-Credential=<credential>
```

La respuesta a esta acción tendrá un aspecto similar al siguiente. Esta respuesta se ha truncado.

```
<DescribeCacheParametersResponse xmlns="http://elasticache.amazonaws.com/
doc/2013-06-15/">
  <DescribeCacheParametersResult>
    <CacheClusterClassSpecificParameters>
      <CacheNodeTypeSpecificParameter>
        <DataType>integer</DataType>
        <Source>system</Source>
        <IsModifiable>false</IsModifiable>
        <Description>The maximum configurable amount of memory to use to store items,
 in megabytes.</Description>
        <CacheNodeTypeSpecificValues>
          <CacheNodeTypeSpecificValue>
            <Value>1000</Value>
            <CacheClusterClass>cache.c1.medium</CacheClusterClass>
          </CacheNodeTypeSpecificValue>
          <CacheNodeTypeSpecificValue>
            <Value>6000</Value>
            <CacheClusterClass>cache.c1.xlarge</CacheClusterClass>
          </CacheNodeTypeSpecificValue>
          <CacheNodeTypeSpecificValue>
            <Value>7100</Value>
            <CacheClusterClass>cache.m1.large</CacheClusterClass>
          </CacheNodeTypeSpecificValue>
          <CacheNodeTypeSpecificValue>
            <Value>1300</Value>
            <CacheClusterClass>cache.m1.small</CacheClusterClass>
          </CacheNodeTypeSpecificValue>

...output omitted...

    </CacheClusterClassSpecificParameters>
  </DescribeCacheParametersResult>
  <ResponseMetadata>
    <RequestId>6d355589-af49-11e0-97f9-279771c4477e</RequestId>
```

```
  </ResponseMetadata>
</DescribeCacheParametersResponse>
```

Para obtener más información, consulte `DescribeCacheParameters`.

Modificación de un grupo de parámetros de ElastiCache

> **Important**
>
> No es posible modificar ningún grupo de parámetros predeterminado.

Es posible modificar algunos parámetros de un grupo de parámetros. Dichos valores de parámetros se aplican a los clústeres asociados al grupo de parámetros. Para obtener más información acerca de cuándo se aplica un cambio de valor de parámetro a un grupo de parámetros, consulte Parámetros de Valkey y Redis OSS y Parámetros específicos de Memcached.

Modificación de un grupo de parámetros (consola)

En el siguiente procedimiento se muestra cómo cambiar el valor del parámetro `cluster-enabled` mediante la consola de ElastiCache. Puede usar el mismo procedimiento para cambiar el valor de cualquier parámetro.

Para cambiar el valor de un parámetro mediante la consola de ElastiCache

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. Para ver una lista de todos los grupos de parámetros disponibles, en el panel de navegación izquierdo, elija Parameter Groups.
3. Elija el grupo de parámetros que desea modificar activando la casilla situada a la izquierda del nombre del grupo de parámetros.

   Los parámetros del grupo de parámetros se mostrarán en la parte inferior de la pantalla. Es posible que necesite desplazarse por las páginas de la lista para ver todos los parámetros.
4. Para modificar uno o varios parámetros, elija Edit Parameters.
5. En la pantalla Edit Parameter Group:, desplácese mediante las flechas izquierda y derecha hasta que encuentre el parámetro `binding_protocol` y, a continuación, escriba `ascii` en la columna Value.

6. Elija Save changes (Guardar cambios).
7. Para Memcached, para encontrar el nombre del parámetro que ha cambiado, consulte Parámetros específicos de Memcached. Si los cambios realizados en el parámetro se aplican tras reiniciar, reinicie todos los clústeres que usen este grupo de parámetros. Para obtener más información, consulte Rebooting clusters.
8. Con Valkey y Redis OSS, para encontrar el nombre del parámetro que ha cambiado, consulte Parámetros de Valkey y Redis OSS. Si tiene un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) y realiza cambios en los siguientes parámetros, debe reiniciar los nodos del clúster:

   - activerehashing
   - bases de datos

   Para obtener más información, consulte Nodos de reinicio.

   > Cambios de parámetros de Valkey o Redis OSS (modo de clúster habilitado)
   >
   > Si realiza cambios en los siguientes parámetros en un clúster de Valkey o Redis OSS (modo de clúster habilitado), realice los siguientes pasos.
   >
   > - activerehashing
   > - bases de datos
   >
   > 1. Con Redis OSS, puede crear una copia de seguridad manual de su clúster. Consulte Copias de seguridad manuales.
   > 2. Eliminar el clúster. Consulte Deleting clusters.
   > 3. Restaure el clúster con la copia de seguridad y el grupo de parámetros modificados para propagar los datos en el nuevo clúster. Consulte Restauración desde una copia de seguridad a una nueva caché.
   >
   > Los cambios en otros parámetros no requieren esta acción.

Modificación de un grupo de parámetros (AWS CLI)

Para cambiar el valor de un parámetro mediante la AWS CLI, use el comando `modify-cache-parameter-group`.

Example

Con Memcached, para encontrar el nombre y los valores permitidos del parámetro que desea cambiar, consulte Parámetros específicos de Memcached.

El siguiente código de muestra establece el valor de dos parámetros, chunk_size y chunk_size_growth_fact, en el grupo de parámetros `myMem14`.

Para Linux, macOS o Unix:

```
aws elasticache modify-cache-parameter-group \
    --cache-parameter-group-name myMem14 \
    --parameter-name-values \
        ParameterName=chunk_size,ParameterValue=96 \
        ParameterName=chunk_size_growth_fact,ParameterValue=1.5
```

Para Windows:

```
aws elasticache modify-cache-parameter-group ^
    --cache-parameter-group-name myMem14 ^
    --parameter-name-values ^
        ParameterName=chunk_size,ParameterValue=96 ^
        ParameterName=chunk_size_growth_fact,ParameterValue=1.5
```

El resultado de este comando será algo similar a lo que se muestra a continuación.

```
{
    "CacheParameterGroupName": "myMem14"
}
```

Example

Con Valkey y Redis OSS, para encontrar el nombre y los valores permitidos del parámetro que desea cambiar, consulte Parámetros de Valkey y Redis OSS.

El siguiente código de muestra establece el valor de dos parámetros, reserved-memory-percent y cluster-enabled, en el grupo de parámetros `myredis32-on-30`. Establecemos reserved-memory-

percent en `30` (30 %) y cluster-enabled en `yes` de modo que el grupo de parámetros se pueda utilizar con clústeres (grupos de replicación) de Valkey o Redis OSS (modo de clúster habilitado).

Para Linux, macOS o Unix:

```
aws elasticache modify-cache-parameter-group \
    --cache-parameter-group-name myredis32-on-30 \
    --parameter-name-values \
        ParameterName=reserved-memory-percent,ParameterValue=30 \
        ParameterName=cluster-enabled,ParameterValue=yes
```

Para Windows:

```
aws elasticache modify-cache-parameter-group ^
    --cache-parameter-group-name myredis32-on-30 ^
    --parameter-name-values ^
        ParameterName=reserved-memory-percent,ParameterValue=30 ^
        ParameterName=cluster-enabled,ParameterValue=yes
```

El resultado de este comando será algo similar a lo que se muestra a continuación.

```
{
    "CacheParameterGroupName": "my-redis32-on-30"
}
```

Para obtener más información, consulte `modify-cache-parameter-group`.

Para encontrar el nombre del parámetro que ha cambiado, consulte Parámetros de Valkey y Redis OSS.

Si tiene un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) y realiza cambios en los siguientes parámetros, debe reiniciar los nodos del clúster:

- activerehashing
- bases de datos

Para obtener más información, consulte Nodos de reinicio.

Cambios de parámetros de Valkey o Redis OSS (modo de clúster habilitado)

Si realiza cambios en los siguientes parámetros en un clúster de Valkey o Redis OSS (modo de clúster habilitado), realice los siguientes pasos.

- activerehashing
- bases de datos

1. Cree una copia de seguridad manual de su clúster. Consulte Copias de seguridad manuales.
2. Eliminar el clúster. Consulte Deleting clusters.
3. Restaure el clúster con la copia de seguridad y el grupo de parámetros modificados para propagar los datos en el nuevo clúster. Consulte Restauración desde una copia de seguridad a una nueva caché.

Los cambios en otros parámetros no requieren esta acción.

Modificación de un grupo de parámetros (API de ElastiCache)

Para cambiar los valores de parámetro de un grupo de parámetros mediante la API de ElastiCache, utilice la acción `ModifyCacheParameterGroup`.

Example

Con Memcached, para encontrar el nombre y los valores permitidos del parámetro que desea cambiar, consulte Parámetros específicos de Memcached.

El siguiente código de muestra establece el valor de dos parámetros, chunk_size y chunk_size_growth_fact, en el grupo de parámetros `myMem14`.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=ModifyCacheParameterGroup
   &CacheParameterGroupName=myMem14
   &ParameterNameValues.member.1.ParameterName=chunk_size
   &ParameterNameValues.member.1.ParameterValue=96
   &ParameterNameValues.member.2.ParameterName=chunk_size_growth_fact
   &ParameterNameValues.member.2.ParameterValue=1.5
```

```
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &Version=2015-02-02
   &X-Amz-Credential=<credential>
```

Example

Con Valkey y Redis OSS, para encontrar el nombre y los valores permitidos del parámetro que desea cambiar, consulte Parámetros de Valkey y Redis OSS.

El siguiente código de muestra establece el valor de dos parámetros, reserved-memory-percent y cluster-enabled, en el grupo de parámetros `myredis32-on-30`. Establecemos reserved-memory-percent en `30` (30 %) y cluster-enabled en `yes` de modo que el grupo de parámetros se pueda utilizar con clústeres (grupos de replicación) de Valkey o Redis OSS (modo de clúster habilitado).

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=ModifyCacheParameterGroup
   &CacheParameterGroupName=myredis32-on-30
   &ParameterNameValues.member.1.ParameterName=reserved-memory-percent
   &ParameterNameValues.member.1.ParameterValue=30
   &ParameterNameValues.member.2.ParameterName=cluster-enabled
   &ParameterNameValues.member.2.ParameterValue=yes
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &Version=2015-02-02
   &X-Amz-Credential=<credential>
```

Para obtener más información, consulte `ModifyCacheParameterGroup`.

Si tiene un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) y realiza cambios en los siguientes parámetros, debe reiniciar los nodos del clúster:

- activerehashing
- bases de datos

Para obtener más información, consulte Reinicio de nodos.

Cambios de parámetros de Valkey o Redis OSS (modo de clúster habilitado)

Si realiza cambios en los siguientes parámetros en un clúster de Valkey o Redis OSS (modo de clúster habilitado), realice los siguientes pasos.

- activerehashing
- bases de datos

1. Cree una copia de seguridad manual de su clúster. Consulte Copias de seguridad manuales.
2. Eliminar el clúster. Consulte Eliminación de un clúster en ElastiCache.
3. Restaure el clúster con la copia de seguridad y el grupo de parámetros modificados para propagar los datos en el nuevo clúster. Consulte Restauración desde una copia de seguridad a una nueva caché.

Los cambios en otros parámetros no requieren esta acción.

Eliminación de un grupo de parámetros de ElastiCache

Puede eliminar un grupo de parámetros personalizados con la consola de ElastiCache, la AWS CLI o la API de ElastiCache.

No podrá eliminar un grupo de parámetros si está asociado a un clúster. Tampoco podrá eliminar ninguno de los grupos de parámetros predeterminados.

Eliminación de un grupo de parámetros (consola)

En el siguiente procedimiento se muestra cómo eliminar un grupo de parámetros mediante la consola de ElastiCache.

Para eliminar un grupo de parámetros con la consola de ElastiCache

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. Para ver una lista de todos los grupos de parámetros disponibles, en el panel de navegación izquierdo, elija Parameter Groups.
3. Elija los grupos de parámetros que desea eliminar activando la casilla situada a la izquierda del nombre del grupo de parámetros.

   El botón Delete se activará.
4. Elija Eliminar.

   Aparecerá la pantalla de confirmación Delete Parameter Groups.
5. Para eliminar los grupos de parámetros, en la pantalla de confirmación Delete Parameter Groups, elija Delete.

   Para conservar los grupos de parámetros, elija Cancel.

Eliminación de un grupo de parámetros (AWS CLI)

Para eliminar un grupo de parámetros mediante la AWS CLI, use el comando `delete-cache-parameter-group`. Para que el grupo de parámetros se elimine, el grupo de parámetros especificado mediante `--cache-parameter-group-name` no puede tener ningún clúster asociado al grupo ni puede ser un grupo de parámetros predeterminado.

El siguiente código de muestra elimina el grupo de parámetros myMem14.

Example

Para Linux, macOS o Unix:

```
aws elasticache delete-cache-parameter-group \
    --cache-parameter-group-name myRed28
```

Para Windows:

```
aws elasticache delete-cache-parameter-group ^
    --cache-parameter-group-name myRed28
```

Para obtener más información, consulte delete-cache-parameter-group.

Eliminación de un grupo de parámetros (API de ElastiCache)

Para eliminar un grupo de parámetros mediante la API de ElastiCache, utilice la acción `DeleteCacheParameterGroup`. Para que el grupo de parámetros se elimine, el grupo de parámetros especificado mediante `CacheParameterGroupName` no puede tener ningún clúster asociado al grupo ni puede ser un grupo de parámetros predeterminado.

Example

Con Memcached, el siguiente código de ejemplo elimina el grupo de parámetros myMem14.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DeleteCacheParameterGroup
   &CacheParameterGroupName=myMem14
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &Version=2015-02-02
   &X-Amz-Credential=<credential>
```

Example

El siguiente código de muestra elimina el grupo de parámetros myRed28.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DeleteCacheParameterGroup
   &CacheParameterGroupName=myRed28
```

```
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

Para obtener más información, consulte DeleteCacheParameterGroup.

Parámetros específicos del motor

Valkey y Redis OSS

La mayoría de los parámetros de Valkey 8 son compatibles con los parámetros de Redis OSS 7.1. Los parámetros de Valkey 7.2 son los mismos que los parámetros de Redis OSS 7.

Si no se especifica ningún grupo de parámetros para el clúster de Valkey o Redis OSS, se usará un grupo de parámetros predeterminado apropiado para la versión del motor. No puede cambiar los valores de los parámetros de un grupo de parámetros predeterminado. Sin embargo, puede crear un grupo de parámetros personalizado y asignarlo a su clúster en cualquier momento, siempre y cuando los valores de los parámetros modificables condicionalmente sean iguales en ambos grupos de parámetros. Para obtener más información, consulte Creación de un grupo de parámetros de ElastiCache.

Temas

- Parámetros de Valkey y Redis OSS
- Parámetros específicos de Memcached

Parámetros de Valkey y Redis OSS

Temas

- Cambios en los parámetros de Valkey 8.2
- Cambios de parámetros de Valkey 8.1
- Cambios de parámetros de Valkey 8.0
- Cambios en los parámetros de Valkey 7.2 y Redis OSS 7
- Cambios de parámetros de Redis OSS 6.x
- Cambios de parámetros de Redis OSS 5.0.3
- Cambios de parámetros de Redis OSS 5.0.0
- Cambios de parámetros de Redis OSS 4.0.10
- Cambios de parámetros de Redis OSS 3.2.10
- Cambios de parámetros de Redis OSS 3.2.6
- Cambios de parámetros de Redis OSS 3.2.4
- Parámetros que se han añadido a Redis OSS 2.8.24 (mejorado)
- Parámetros que se han añadido a Redis OSS 2.8.23 (mejorado)
- Parámetros que se han añadido a Redis OSS 2.8.22 (mejorado)
- Parámetros que se han añadido a Redis OSS 2.8.21
- Parámetros que se han añadido a Redis OSS 2.8.19
- Parámetros que se han añadido a Redis OSS 2.8.6
- Parámetros de Redis OSS 2.6.13
- Parámetros específicos de tipos de nodos de Redis OSS

Cambios en los parámetros de Valkey 8.2

Familia de grupos de parámetros: valkey8

> **Note**
>
> - Los cambios en los parámetros de Valkey 8.2 no se aplican a Valkey 8.1
> - Los grupos de parámetros de Valkey 8.0 y superiores son incompatibles con Redis OSS 7.2.4.

- En Valkey 8.2, los siguientes comandos no están disponibles para las cachés sin servidor: `commandlog`, `commandlog get`, `commandlog help`, `commandlog len` y `commandlog reset`.

Nuevos grupos de parámetros en Valkey 8.2

Nombre	Detalles	Descripción
search-fanout-target-mode (añadido en 8.2)	Predeterminado: cliente Tipo: cadena Modificable: sí Aplicación de los cambios: inmediatamente	El parámetro de configuración search-fanout-target-mode controla cómo se distribuyen las consultas de búsqueda entre los nodos de un entorno de clústeres de Valkey. Esta configuración acepta dos valores: “throughput”, que optimiza el rendimiento máximo mediante la distribución aleatoria de las consultas de búsqueda en todos los nodos del clúster, independientemente del tipo de cliente o del estado READONLY, y “client”, que respeta las características de la conexión del cliente al enrutar los clientes que no son READONLY solo hacia los nodos principales, los clientes READONLY en las conexiones de réplica solo hacia los nodos de réplica y los clientes READONLY en las conexiones principales de

Nombre	Detalles	Descripción
		forma aleatoria hacia todos los nodos. El comportamiento predeterminado es el modo “client”, lo que significa que el sistema respetará los tipos de conexión de cliente y el estado READONLY para las decisiones de enrutamiento de consultas. Utilice el modo “throughput” para las cargas de trabajo de búsqueda de gran volumen en las que quiera aprovechar al máximo los recursos del clúster, y el modo “client” cuando desee mantener la separación entre lectura y escritura y respetar los patrones de conexión READONLY en la aplicación.

Nombre	Detalles	Descripción
search-default-timeout-ms	Predeterminado: 50000 Valores permitidos: de 1 a 60000 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Es el tiempo de espera predeterminado de la consulta de búsqueda de Valkey (en milisegundos).
search-enable-partial-results	Valor predeterminado: yes Valores permitidos: yes, no Tipo: booleano Modificable: sí Aplicación de los cambios: inmediatamente	Configura el comportamiento de error de las consultas en la búsqueda de Valkey. Si está habilitado, las consultas de búsqueda devuelven resultados parciales si se agota el tiempo de espera en una o más particiones. Si está deshabilitado, la partición del tiempo de espera provocará el fallo de toda la consulta de búsqueda y se generará un error.

Cambios de parámetros de Valkey 8.1

Familia de grupos de parámetros: valkey8

Note

- Los cambios en los parámetros de Valkey 8.1 no se aplican a Valkey 8.0
- Los grupos de parámetros de Valkey 8.0 y superiores son incompatibles con Redis OSS 7.2.4.
- En Valkey 8.1, los siguientes comandos no están disponibles para las cachés sin servidor: `commandlog`, `commandlog get`, `commandlog help`, `commandlog len` y `commandlog reset`.

Nuevos grupos de parámetros en Valkey 8.1

Nombre	Detalles	Descripción
commandlog-large-request-max-len (añadido en 8.1)	Predeterminado: 1048576 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Tamaño máximo, en bytes, de las solicitudes que debe registrar la característica Registro de comandos de Valkey.
commandlog-large-request-max-len (añadido en 8.1)	Valor predeterminado: 128 Valores permitidos: 0-1024 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Longitud máxima de la característica Registro de comandos de Valkey para las solicitudes.
commandlog-reply-larger-than (añadido en 8.1)	Predeterminado: 1048576	Tamaño máximo, en bytes, de las respuestas que debe

Nombre	Detalles	Descripción
	Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	registrar la característica Registro de comandos de Valkey.
commandlog-large-reply-max-len (añadido en 8.1)	Valor predeterminado: 128 Valores permitidos: 0-1024 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Tamaño máximo de la característica Registro de comandos de Valkey para las respuestas.

Cambios de parámetros de Valkey 8.0

Familia de grupos de parámetros: valkey8

> **Note**
>
> Redis OSS 7.2.4 es incompatibles con los grupos de parámetros de Valkey 8.0 y versiones posteriores.

Cambios de parámetros específicos en Valkey 8.0

Nombre	Detalles	Descripción
repl-backlog-size	Predeterminado: 10485760 Tipo: número entero Modificable: sí	Tamaño, en bytes, del búfer de tareas pendientes del nodo principal. Las tareas pendientes se usan para registrar actualizaciones de

Nombre	Detalles	Descripción
	Aplicación de los cambios: inmediatamente	los datos en el nodo principal . Cuando una réplica de lectura se conecta al nodo principal, intenta realizar una sincronización parcial (psync), donde se aplican los datos de las tareas pendientes para ponerse al día con el nodo principal. Si se produce un error en la psync, se debe realizar una sincronización completa. El valor mínimo para este parámetro es 16384. Nota: A partir de Redis OSS 2.8.22, este parámetro se aplica al clúster principal , así como a las réplicas de lectura.
maxmemory-samples	Valor predeterminado: 3 Valores permitidos: de 1 a 64 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Para los cálculos de elementos menos usados recientemente (LRU) y de tiempo de vida (TTL), este parámetro represent a el tamaño de muestra de las claves que se deben comprobar. De forma predeterminada, Redis OSS elige 3 claves y utiliza la que se usó menos recientemente.

Nuevos grupos de parámetros en Valkey 8.0

Nombre	Detalles	Descripción

extended-redis-compatibility	Valores permitidos: yes, no
	
	Valor predeterminado: yes
	
	Tipo: booleano
	
	Modificable: sí
	
	Aplicación de los cambios: inmediatamente
		El modo de compatibilidad extendido con Redis OSS hace que Valkey se haga pasar por Redis OSS 7.2. Active esta opción solo si tiene problemas con las herramien tas o los clientes.
		
		Efectos de cara al cliente:
		
		• `LOADING`: Redis OSS está cargando el conjunto de datos en la memoria
		• `BUSY`: Redis OSS está ocupado
		• `MISCONF`: Redis OSS se configura de una de las siguientes maneras:
		  • El comando `HELLO` devuelve “server” => “redis” y “version” => “7.2.4” (nuestra versión compatible con Redis OSS).
		  • El campo `INFO` del modo se denomina “redis_mo de”.

Grupos de parámetros retirados de Valkey 8.0

Nombre	Detalles	Descripción
lazyfree-lazy-eviction	Valores permitidos: yes, no Valor predeterminado: no Tipo: booleano Modificable: sí Aplicación de los cambios: inmediatamente	Realiza una eliminación asíncrona en las expulsiones.
lazyfree-lazy-expire	Valores permitidos: yes, no Valor predeterminado: no Tipo: booleano Modificable: sí Aplicación de los cambios: inmediatamente	Realiza una eliminación asíncrona en las claves vencidas.
lazyfree-lazy-server-del	Valores permitidos: yes, no Valor predeterminado: no Tipo: booleano Modificable: sí Aplicación de los cambios: inmediatamente	Realiza una eliminación asíncrona de los comandos que actualizan valores.
lazyfree-lazy-user-del	Valor predeterminado: no Tipo: String	Cuando el valor se establece en “yes”, el comando DEL

Nombre	Detalles	Descripción
	Modificable: sí Los cambios surten efecto: inmediatamente en todos los clústeres	actúa de la misma manera que UNLINK.
replica-lazy-flush	Valor predeterminado: yes Tipo: booleano Modificable: no Nombre anterior: slave-lazy-flush	Realiza un volcado de base de datos asíncrono durante la sincronización de réplicas.

Cambios en los parámetros de Valkey 7.2 y Redis OSS 7

Familia de grupos de parámetros: valkey7

Los grupos de parámetros predeterminados de Valkey 7.2 son los siguientes:

- `default.valkey7`: utilice este grupo de parámetros, o uno derivado de él, para clústeres y grupos de replicación de Valkey (modo de clúster deshabilitado).
- `default.valkey7.cluster.on`: utilice este grupo de parámetros, o uno derivado de él, para clústeres y grupos de replicación de Valkey (modo de clúster habilitado).

Familia del grupo de parámetros: redis7

Los grupos de parámetros predeterminados de Redis OSS 7 son los siguientes:

- `default.redis7`: utilice este grupo de parámetros, o uno derivado de él, para clústeres y grupos de replicación de Redis OSS (modo de clúster deshabilitado).
- `default.redis7.cluster.on`: utilice este grupo de parámetros, o uno derivado de él, para clústeres y grupos de replicación de Redis OSS (modo de clúster habilitado).

Cambios de parámetros específicos

Los parámetros que se han añadido en Redis OSS 7 son los siguientes. Valkey 7.2 también admite estos parámetros.

Nombre	Detalles	Descripción
`cluster-allow-pubsubshard-when-down`	Valores permitidos: `yes`, `no` Valor predeterminado: `yes` Tipo: cadena Modificable: sí Los cambios surten efecto: inmediatamente en todos los nodos del clúster.	Cuando se establece sí como valor predeterminado, se permite a los nodos atender el tráfico de particiones de pubsub mientras el clúster esté inactivo, siempre y cuando crea que es el propietario de las ranuras.
`cluster-preferred-endpoint-type`	Valores permitidos: `ip`, `tls-dynamic` Valor predeterminado: `tls-dynamic` Tipo: cadena Modificable: sí Los cambios surten efecto: inmediatamente en todos los nodos del clúster.	Este valor controla el punto de conexión que se devuelve para las solicitudes MOVED/ASKING, así como el campo de punto de conexión para `CLUSTER SLOTS` y `CLUSTER SHARDS`. Cuando el valor se establece en IP, el nodo comunicará su dirección IP. Cuando el valor se establece en tls-dynamic, el nodo comunicará un nombre de host cuando esté habilitado el cifrado en tránsito y, en caso contrario, una dirección IP.
`latency-tracking`	Valores permitidos: `yes`, `no` Valor predeterminado: `no` Tipo: cadena Modificable: sí	Cuando se establece en sí, realiza un seguimiento de las latencias por comando y permite exportar la distribución de percentiles mediante el comando de estadísticas de latencia `INFO` y las distribuciones de latencia acumulada (histogramas) mediante el comando `LATENCY`.

Nombre	Detalles	Descripción
	Los cambios surten efecto: inmediatamente en todos los nodos del clúster.	
`hash-max-listpack-entries`	Valores permitidos: `0+` Valor predeterminado: `512` Tipo: número entero Modificable: sí Los cambios surten efecto: inmediatamente en todos los nodos del clúster.	El número máximo de entradas de hash para comprimir el conjunto de datos.
`hash-max-listpack-value`	Valores permitidos: `0+` Valor predeterminado: `64` Tipo: número entero Modificable: sí Los cambios surten efecto: inmediatamente en todos los nodos del clúster.	El umbral de entradas de hash más grandes para comprimir el conjunto de datos.

Nombre	Detalles	Descripción
`zset-max-listpack-entries`	Valores permitidos: `0+` Valor predeterminado: `128` Tipo: número entero Modificable: sí Los cambios surten efecto: inmediatamente en todos los nodos del clúster.	El número máximo de entradas de conjuntos ordenados para comprimir el conjunto de datos.
`zset-max-listpack-value`	Valores permitidos: `0+` Valor predeterminado: `64` Tipo: número entero Modificable: sí Los cambios surten efecto: inmediatamente en todos los nodos del clúster.	El umbral de entradas de conjuntos ordenados más grandes para comprimir el conjunto de datos.

Los parámetros cambiados en Redis OSS 7 son los siguientes.

Nombre	Detalles	Descripción
`activerehashing`	Modificable: no. En Redis OSS 7, este parámetro está oculto y habilitado de forma predeterminada. Para desactivarlo, debe crear un caso de soporte.	Modificable era sí.

Los parámetros eliminados de Redis OSS 7 son los siguientes.

Nombre	Detalles	Descripción
`hash-max-ziplist-entries`	Valores permitidos: `0+` Valor predeterminado: `512` Tipo: número entero Modificable: sí Los cambios surten efecto: inmediatamente en todos los nodos del clúster.	Use `listpack` en lugar de `ziplist` para representar una codificación de hash pequeña
`hash-max-ziplist-value`	Valores permitidos: `0+` Valor predeterminado: `64` Tipo: número entero Modificable: sí Los cambios surten efecto: inmediatamente en todos los nodos del clúster.	Use `listpack` en lugar de `ziplist` para representar una codificación de hash pequeña
`zset-max-ziplist-entries`	Valores permitidos: `0+` Valor predeterminado: `128` Tipo: número entero Modificable: sí Los cambios surten efecto: inmediatamente en todos los nodos del clúster.	Use `listpack` en lugar de `ziplist` para representar una codificación de hash pequeña.

Nombre	Detalles	Descripción
`zset-max-ziplist-value`	Valores permitidos: `0+` Valor predeterminado: `64` Tipo: número entero Modificable: sí Los cambios surten efecto: inmediatamente en todos los nodos del clúster.	Use `listpack` en lugar de `ziplist` para representar una codificación de hash pequeña.
`list-max-ziplist-size`	Valores permitidos: Valor predeterminado: `-2` Tipo: número entero Modificable: sí Los cambios surten efecto: inmediatamente en todos los nodos del clúster.	El número de entradas permitido por nodo de lista interna.

Cambios de parámetros de Redis OSS 6.x

Familia del grupo de parámetros: redis6.x

Los grupos de parámetros predeterminados de Redis OSS 6.x son los siguientes:

- `default.redis6.x`: utilice este grupo de parámetros, o uno derivado de él, para clústeres y grupos de replicación de Valkey o Redis OSS (modo de clúster deshabilitado).
- `default.redis6.x.cluster.on`: utilice este grupo de parámetros, o uno derivado de él, para clústeres y grupos de replicación de Valkey o Redis OSS (modo de clúster habilitado).

> **Note**
>
> En la versión 6.2 del motor de Redis OSS, cuando se introdujo la familia de nodos r6gd para usarla con Organización de datos en niveles en ElastiCache, solo se admitían las políticas de memoria máxima noeviction, volatile-lru y allkeys-lru con los tipos de nodo r6gd.

Para obtener más información, consulte Versión 6.2 de ElastiCache para Redis OSS (mejorada) y Versión 6.0 de ElastiCache para Redis OSS (mejorada).

Los parámetros que se han añadido en Redis OSS 6.x son los siguientes.

Detalles	Descripción	
`acl-pubsub-default (added in 6.2)`	Valores permitidos: `resetchannels ` , `allchannels` Valor predeterminado: `allchannels` Tipo: cadena Modificable: sí Los cambios surten efecto: los usuarios de Redis OSS existentes asociados al clúster seguirán teniendo los permisos existentes. Actualice los usuarios o reinicie el clúster para actualizar los usuarios de Redis OSS existentes.	Permisos de canal pubsub predeterminados para los usuarios de ACL implementados en este clúster.
`cluster-allow-reads-when-do`	Valor predeterminado: no Tipo: String	Cuando se establece en Sí, un grupo de replicación de Redis OSS (modo de clúster habilitado) continúa procesando comandos

Detalles	Descripción	
`wn (added in 6.0)`	Modificable: sí Los cambios surten efecto: inmediatamente en todos los clústeres	de lectura incluso cuando un nodo no puede alcanzar un quórum de nodos principales. Cuando se establece en el valor predeterminado no, el grupo de reproducción rechaza todos los comandos. Se recomienda establecer este valor en yes (sí) si utiliza un clúster con menos de tres grupos de nodos o si la aplicación puede manejar de forma segura las lecturas obsoletas.
`tracking-table-max-keys (added in 6.0)`	Predeterminado: 1 000 000 Tipo: número Modificable: sí Los cambios surten efecto: inmediatamente en todos los clústeres	Para ayudar al almacenamiento en caché en el cliente, Redis OSS permite realizar un seguimiento de los clientes que han accedido a las claves. Cuando se modifica la clave rastreada, se envían mensajes de invalidación a todos los clientes para notificarles que sus valores almacenados en caché ya no son válidos. Este valor permite especificar el límite superior de esta tabla. Una vez superado este valor de parámetro, los clientes se envían invalidación aleatoriamente. Este valor debe ajustarse para limitar la utilización de memoria sin dejar de realizar un seguimiento de las claves suficientes. Las claves también se invalidan en condiciones de poca memoria.

Detalles	Descripción	
`acllog-max-len (added in 6.0)`	Valor predeterminado: 128 Tipo: número Modificable: sí Los cambios surten efecto: inmediatamente en todos los clústeres	Este valor corresponde al número máximo de entradas en el registro ACL.
`active-expire-effort (added in 6.0)`	Valor predeterminado: 1 Tipo: número Modificable: sí Los cambios surten efecto: inmediatamente en todos los clústeres	Redis OSS elimina las claves que han superado su periodo de vida por medio de dos mecanismos. En uno, se accede a una clave y se encuentra que ha vencido. En el otro, un trabajo periódico muestra claves y hace que se venzan aquellas que han excedido su periodo de vida. Este parámetro define la cantidad de esfuerzo que Redis OSS utiliza para hacer que caduquen elementos en el trabajo periódico. El valor predeterminado de 1 intenta evitar tener más del 10 % de las claves vencidas que todavía se encuentran en la memoria. También intenta evitar consumir más del 25 % de la memoria total y agregar latencia al sistema. Puede aumentar este valor hasta 10 para aumentar la cantidad de esfuerzo invertido en las claves vencidas. La desventaja es mayor CPU y una latencia potencialmente mayor. Recomendamos un valor de 1 a menos que vea un uso elevado de memoria y pueda tolerar un aumento en la utilización de la CPU.

Detalles	Descripción	
`lazyfree-lazy-user-del (added in 6.0)`	Valor predeterminado: no Tipo: String Modificable: sí Los cambios surten efecto: inmediatamente en todos los clústeres	Cuando el valor se establece en yes (sí), el comando `DEL` actúa de la misma manera que `UNLINK`.

Los parámetros que se han eliminado de Redis OSS 6.x son los siguientes.

Nombre	Detalles	Descripción
`lua-replicate-commands`	Valores permitidos: sí/no Valor predeterminado: yes Tipo: booleano Modificable: sí Aplicación de los cambios: inmediatamente	Indica si siempre se debe habilitar o no la replicación de los efectos de Lua en los scripts de Lua.

Cambios de parámetros de Redis OSS 5.0.3

Familia del grupo de parámetros: redis5.0

Grupos de parámetros predeterminados de Redis OSS 5.0

- `default.redis5.0`: utilice este grupo de parámetros, o uno derivado de él, para clústeres y grupos de replicación de Valkey o Redis OSS (modo de clúster deshabilitado).
- `default.redis5.0.cluster.on`: utilice este grupo de parámetros, o uno derivado de él, para clústeres y grupos de replicación de Valkey o Redis OSS (modo de clúster habilitado).

Parámetros que se han añadido en Redis OSS 5.0.3

Nombre	Detalles	Descripción
`rename-commands`	Predeterminado: none Tipo: String Modificable: sí Los cambios surten efecto: inmediatamente en todos los clústeres	Una lista separada por espacios de los comandos de Redis OSS a los que se ha cambiado el nombre. A continuación se muestra una lista restringida de comandos disponibles a lo que se puede cambiar el nombre: `APPEND AUTH BITCOUNT BITFIELD BITOP BITPOS BLPOP BRPOP BRPOPLPUSH BZPOPMIN BZPOPMAX CLIENT CLUSTER COMMAND DBSIZE DECR DECRBY DEL DISCARD DUMP ECHO EVAL EVALSHA EXEC EXISTS EXPIRE EXPIREAT FLUSHALL FLUSHDB GEOADD GEOHASH GEOPOS GEODIST GEORADIUS GEORADIUSBYMEMBER GET GETBIT GETRANGE GETSET HDEL HEXISTS HGET HGETALL HINCRBY HINCRBYFLOAT HKEYS HLEN HMGET HMSET HSET HSETNX HSTRLEN HVALS INCR INCRBY INCRBYFLOAT INFO KEYS LASTSAVE LINDEX LINSERT LLEN LPOP LPUSH LPUSHX LRANGE LREM LSET LTRIM MEMORY MGET MONITOR MOVE MSET MSETNX MULTI OBJECT PERSIST PEXPIRE PEXPIREAT PFADD PFCOUNT PFMERGE PING PSETEX PSUBSCRIBE PUBSUB PTTL PUBLISH PUNSUBSCRIBE RANDOMKEY READONLY READWRITE RENAME RENAMENX RESTORE ROLE RPOP RPOPLPUSH RPUSH RPUSHX SADD SCARD SCRIPT SDIFF SDIFFSTORE SELECT SET SETBIT SETEX SETNX SETRANGE`

Nombre	Detalles	Descripción
		SINTER SINTERSTORE SISMEMBER SLOWLOG SMEMBERS SMOVE SORT SPOP SRANDMEMBER SREM STRLEN SUBSCRIBE SUNION SUNIONSTORE SWAPDB TIME TOUCH TTL TYPE UNSUBSCRIBE UNLINK UNWATCH WAIT WATCH ZADD ZCARD ZCOUNT ZINCRBY ZINTERSTORE ZLEXCOUNT ZPOPMAX ZPOPMIN ZRANGE ZRANGEBYLEX ZREVRANGEBYLEX ZRANGEBYSCORE ZRANK ZREM ZREMRANGEBYLEX ZREMRANGEBYRANK ZREMRANGEBYSCORE ZREVRANGE ZREVRANGEBYSCORE ZREVRANK ZSCORE ZUNIONSTORE SCAN SSCAN HSCAN ZSCAN XINFO XADD XTRIM XDEL XRANGE XREVRANGE XLEN XREAD XGROUP XREADGROUP XACK XCLAIM XPENDING GEORADIUS_RO GEORADIUSBYMEMBER_RO LOLWUT XSETID SUBSTR

Para obtener más información, consulte Versión 5.0.6 de ElastiCache para Redis OSS (mejorada).

Cambios de parámetros de Redis OSS 5.0.0

Familia del grupo de parámetros: redis5.0

Grupos de parámetros predeterminados de Redis OSS 5.0

- `default.redis5.0`: utilice este grupo de parámetros, o uno derivado de él, para clústeres y grupos de replicación de Valkey o Redis OSS (modo de clúster deshabilitado).
- `default.redis5.0.cluster.on`: utilice este grupo de parámetros, o uno derivado de él, para clústeres y grupos de replicación de Valkey o Redis OSS (modo de clúster habilitado).

Parámetros que se han añadido en Redis OSS 5.0

Nombre	Detalles	Descripción
`stream-node-max-bytes`	Valores permitidos: 0+ Predeterminado: 4096 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	La estructura de datos de secuencia es un árbol de prefijos de nodos que contiene varios elementos. Utilice esta configuración para especificar el tamaño máximo de un único nodo de un árbol de prefijos in bytes. Si se establece en 0, el tamaño del nodo del árbol es ilimitado.
`stream-node-max-entries`	Valores permitidos: 0+ Predeterminado: 100 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	La estructura de datos de secuencia es un árbol de prefijos de nodos que contiene varios elementos. Utilice esta configuración para especificar el número máximo de elementos que puede contener un único nodo antes de cambiar a un nodo nuevo al agregar entradas nuevas de secuencia. Si se establece en 0, el número de elementos del nodo del árbol es ilimitado.
`active-defrag-max-scan-fields`	Valores permitidos: de 1 a 1000000 Predeterminado: 1000 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Número máximo de campos set/hash/zset/list que se procesarán desde el análisis del diccionario principal.

Nombre	Detalles	Descripción
`lua-replicate-commands`	Valores permitidos: sí/no Valor predeterminado: yes Tipo: booleano Modificable: sí Aplicación de los cambios: inmediatamente	Indica si siempre se debe habilitar o no la replicación de los efectos de Lua en los scripts de Lua.
`replica-ignore-maxmemory`	Valor predeterminado: yes Tipo: booleano Modificable: no	Determina si la réplica no tiene en cuenta el ajuste `maxmemory` al no expulsar elementos independientes del nodo primario

Se ha cambiado el nombre de varios parámetros de Redis OSS en la versión 5.0 del motor en respuesta a los comentarios de la comunidad. Para obtener más información, consulte What's New in Redis OSS 5? En la tabla siguiente, se muestran los nombres nuevos y a qué comandos corresponden en las versiones anteriores.

Parámetros cuyo nombre ha cambiado en Redis OSS 5.0

Nombre	Detalles	Descripción
`replica-lazy-flush`	Valor predeterminado: yes Tipo: booleano Modificable: no Nombre anterior: slave-lazy-flush	Realiza un volcado de base de datos asíncrono durante la sincronización de réplicas.

Nombre	Detalles	Descripción
`client-output-buffer-limit-replica-hard-limit`	Valor predeterminado: para obtener información acerca de los valores, consulte Parámetros específicos de tipos de nodos de Redis OSS Tipo: número entero Modificable: no Nombre anterior: client-output-buffer-limit-slave-hard-limit	Para réplicas de lectura de Redis OSS: si el búfer de salida de un cliente alcanza el número de bytes especificado, el cliente se desconectará.
`client-output-buffer-limit-replica-soft-limit`	Valor predeterminado: para obtener información acerca de los valores, consulte Parámetros específicos de tipos de nodos de Redis OSS Tipo: número entero Modificable: no Nombre anterior: client-output-buffer-limit-slave-soft-limit	Para réplicas de lectura de Redis OSS: si el búfer de salida de un cliente alcanza el número de bytes especificado, el cliente se desconectará solo si esta condición se mantiene durante `client-output-buffer-limit-replica-soft-seconds` .
`client-output-buffer-limit-replica-soft-seconds`	Predeterminado: 60 Tipo: número entero Modificable: no Nombre anterior: client-output-buffer-limit-slave-soft-seconds	Para réplicas de lectura de Redis OSS: si el búfer de salida de un cliente permanece en `client-output-buffer-limit-replica-soft-limit` bytes durante un tiempo superior a esta cantidad de segundos, el cliente se desconectará.

Nombre	Detalles	Descripción
`replica-allow-chaining`	Valor predeterminado: no Tipo: String Modificable: no Nombre anterior: slave-allow-chaining	Determina si una réplica de lectura de Redis OSS puede tener réplicas de lectura propias.
`min-replicas-to-write`	Predeterminado: 0 Tipo: número entero Modificable: sí Nombre anterior: min-slaves-to-write Aplicación de los cambios: inmediatamente	Número mínimo de réplicas de lectura que deben estar disponibles para el nodo principal acepte operaciones de escritura de los clientes. Si el número de réplicas disponibles cae por debajo de este número, el nodo principal dejará de aceptar solicitudes de escritura. Si este parámetro o min-replicas-max-lag es igual a 0, el nodo principal siempre aceptará solicitudes de escritura, incluso cuando no haya réplicas disponibles.
`min-replicas-max-lag`	Valor predeterminado: 10 Tipo: número entero Modificable: sí Nombre anterior: min-slaves-max-lag Aplicación de los cambios: inmediatamente	Número de segundos en los que el nodo principal debe recibir una solicitud de ping desde una réplica de lectura. Si este periodo de tiempo transcurre y el nodo principal no recibe ningún ping, la réplica deja de considerarse como disponible. Si el número de réplicas disponibles cae por debajo del valor min-replicas-to-write, el nodo principal dejará de aceptar operaciones de escritura. Si este parámetro o min-replicas-to-write es iguales a 0, el nodo principal siempre aceptará solicitudes de escritura, incluso cuando no haya réplicas disponibles.

Nombre	Detalles	Descripción
`close-on-replica-write`	Valor predeterminado: yes Tipo: booleano Modificable: sí Nombre anterior: close-on-slave-write Aplicación de los cambios: inmediatamente	Si está habilitado, los clientes que intenten escribir en una réplica de solo lectura se desconectarán.

Parámetros que se han eliminado de Redis OSS 5.0

Nombre	Detalles	Descripción
`repl-timeout`	Predeterminado: 60 Modificable: no	El parámetro no está disponible en esta versión.

Cambios de parámetros de Redis OSS 4.0.10

Familia del grupo de parámetros: redis4.0

Grupos de parámetros predeterminados de Redis OSS 4.0.x

- `default.redis4.0`: utilice este grupo de parámetros, o uno derivado de él, para clústeres y grupos de replicación de Valkey o Redis OSS (modo de clúster deshabilitado).
- `default.redis4.0.cluster.on`: utilice este grupo de parámetros, o uno derivado de él, para clústeres y grupos de replicación de Valkey o Redis OSS (modo de clúster habilitado).

Parámetros que se han cambiado en Redis OSS 4.0.10

Nombre	Detalles	Descripción
`maxmemory-policy`	Valores permitidos: `allkeys-lru` , `volatile-lru` , **`allkeys-lfu`** , **`volatile-lfu`** , `allkeys-random` , `volatile-random` , `volatile-ttl` , `noeviction` Valor predeterminado: volatile-lru Tipo: String Modificable: sí Aplicación de los cambios: inmediatamente	`maxmemory-policy` se agregó en la versión 2.6.13. En la versión 4.0.10, se agregaron dos nuevos valores permitidos: `allkeys-lfu` , que expulsará claves utilizando un criterio LFU aproximado, y `volatile-lfu` , que realizará la expulsión utilizando un criterio de LFU aproximado entre las claves que tienen un vencimiento establecido. En la versión 6.2, cuando se introdujo la familia de nodos r6gd para utilizarla con la organización de datos en niveles, solo las políticas max-memory `noeviction` , `volatile-lru` y `allkeys-lru` se admiten con tipos de nodos r6gd.

Parámetros que se han añadido en Redis OSS 4.0.10

Nombre	Detalles	Descripción
Parámetros de eliminación asíncrona		
`lazyfree-lazy-eviction`	Valores permitidos: sí/no Valor predeterminado: no Tipo: booleano Modificable: sí Aplicación de los cambios: inmediatamente	Performs an asynchronous delete on evictions.

Nombre	Detalles	Descripción
`lazyfree-lazy-expire`	Valores permitidos: sí/no Valor predeterminado: no Tipo: booleano Modificable: sí Aplicación de los cambios: inmediatamente	Performs an asynchronous delete on expired keys.
`lazyfree-lazy-server-del`	Valores permitidos: sí/no Valor predeterminado: no Tipo: booleano Modificable: sí Aplicación de los cambios: inmediatamente	Performs an asynchronous delete for commands which update values.
`slave-lazy-flush`	Valores permitidos: N/A Valor predeterminado: no Tipo: booleano Modificable: no Changes take place: N/A	Performs an asynchronous flushDB during slave sync.

Parámetros de LFU

Nombre	Detalles	Descripción
`lfu-log-factor`	Valores permitidos: cualquier número entero > 0 Valor predeterminado: 10 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Set the log factor, which determines the number of key hits to saturate the key counter.
`lfu-decay-time`	Valores permitidos: cualquier número entero Valor predeterminado: 1 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	The amount of time in minutes to decrement the key counter.

Parámetros de desfragmentación activa

Nombre	Detalles	Descripción
`activedefrag`	Valores permitidos: sí/no Valor predeterminado: no Tipo: booleano Modificable: sí Aplicación de los cambios: inmediatamente	Admite la desfragmentación activa. **Note** En las versiones 7.0 y posteriores de Valkey y Redis OSS, AWS puede realizar automáticamente la desfragmentación cuando sea necesario desde el punto de vista operativo, independientemente de esta configuración.
`active-defrag-ignore-bytes`	Valores permitidos: 10485760-104857600 Predeterminado: 104857600 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Minimum amount of fragmentation waste to start active defrag.

Nombre	Detalles	Descripción
`active-defrag-threshold-lower`	Valores permitidos: 1-100 Valor predeterminado: 10 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Minimum percentage of fragmentation to start active defrag.
`active-defrag-threshold-upper`	Valores permitidos: 1-100 Predeterminado: 100 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Maximum percentage of fragmentation at which we use maximum effort.
`active-defrag-cycle-min`	Valores permitidos: 1-75 Predeterminado: 25 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Minimal effort for defrag in CPU percentage.

Nombre	Detalles	Descripción
`active-defrag-cycle-max`	Valores permitidos: 1-75 Predeterminado: 75 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Maximal effort for defrag in CPU percentage.
Parámetros de búfer de salida de cliente		
`client-query-buffer-limit`	Valores permitidos: 1048576-1073741824 Predeterminado: 1073741824 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Max size of a single client query buffer.
`proto-max-bulk-len`	Valores permitidos: 1048576-536870912 Predeterminado: 536870912 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Max size of a single element request.

Cambios de parámetros de Redis OSS 3.2.10

Familia del grupo de parámetros: redis3.2

En ElastiCache para Redis 3.2.10 no se admiten parámetros adicionales.

Cambios de parámetros de Redis OSS 3.2.6

Familia del grupo de parámetros: redis3.2

En Redis OSS 3.2.6 no se admiten parámetros adicionales.

Cambios de parámetros de Redis OSS 3.2.4

Familia del grupo de parámetros: redis3.2

A partir de Redis OSS 3.2.4, existen dos grupos de parámetros predeterminados.

- `default.redis3.2`: si ejecuta Redis OSS 3.2.4, especifique este grupo de parámetros, o uno derivado de este, si desea crear un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) y seguir utilizando las características adicionales de Redis OSS 3.2.4.
- `default.redis3.2.cluster.on`: especifique este grupo de parámetros, o uno derivado de este, si desea crear un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado).

Temas

- Parámetros nuevos para Redis OSS 3.2.4
- Parámetros cambiados en Redis OSS 3.2.4 (mejorado)

Parámetros nuevos para Redis OSS 3.2.4

Familia del grupo de parámetros: redis3.2

En Redis OSS 3.2.4 se admiten los siguientes parámetros adicionales.

Nombre	Detalles	Descripción
`list-max-ziplist-size`	Valor predeterminado: -2 Tipo: número entero	Las listas se codifican de una manera especial para ahorrar espacio. El número de entradas permitidas por nodo de lista interna se puede especificar como de tamaño máximo fijo o

Nombre	Detalles	Descripción
	Modificable: no	como un número máximo de elementos. Para establecer un tamaño máximo fijo, use un valor de -5 a -1 teniendo en cuenta lo siguiente: • -5: tamaño máximo de 64 Kb; no se recomienda esta opción para cargas de trabajo normales • -4: tamaño máximo de 32 Kb; no se recomienda esta opción • -3: tamaño máximo de 16 Kb; no se recomienda esta opción • -2: tamaño máximo de 8 Kb; se recomienda esta opción • -1: tamaño máximo de 4 Kb; se recomienda esta opción • El uso de un número positivo indican que se puede almacenar hasta dicho número de elementos por cada nodo de lista.

Nombre	Detalles	Descripción
`list-compress-depth`	Predeterminado: 0 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Las listas también pueden comprimirse. La profundidad de compresión es el número de nodos de listas comprimidas de listas rápidas de ambos lados de la lista que se excluirán de la compresión. El principio y el final de la lista están siempre sin comprimir para agilizar las operaciones de inserción y extracción. Los valores son los siguientes: • 0: deshabilitar toda la compresión. • 1: comenzar a comprimir con el primer nodo a partir del principio y el final. [principio]->nodo->nodo->...->nodo->[final] Se comprimen todos los nodos excepto los nodos [principio] y [final]. • 2: comenzar a comprimir con el segundo nodo a partir del principio y el final. [principio]->[siguiente]->nodo->nodo->...->nodo->[penúltimo]->[final] Los nodos [principio], [siguiente], [penúltimo] y [final] no se comprimen. Todos los demás nodos se comprimen. • etc.

Nombre	Detalles	Descripción
`cluster-enabled`	Valor predeterminado: no/yes * Tipo: cadena Modificable: no	Indica si se trata de un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado) en el modo de clúster (sí) o de un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado) en el modo sin clúster (no). Los grupos de replicación de Valkey o Redis OSS (modo de clúster habilitado) en el modo de clúster pueden particionar sus datos en hasta 500 grupos de nodos. * Redis OSS 3.2.x tiene dos grupos de parámetros predeterminados. • `default.redis3.2` – valor predeterminado no. • `default.redis3.2.cluster.on` – valor predeterminado `yes`. .

Nombre	Detalles	Descripción
`cluster-require-full-coverage`	Valor predeterminado: no Tipo: booleano Modificable: sí Aplicación de los cambios: inmediatamente	Cuando se establece en `yes`, los nodos de Valkey o Redis OSS (modo de clúster habilitado) en el modo de clúster dejan de aceptar consultas si detectan que hay al menos una ranura de hash sin cubrir (sin ningún nodo disponible que la atienda). De esta forma, si el clúster está parcialmente inactivo, este deja de estar disponible. El clúster vuelve a estar disponible automáticamente en el momento en que todas las ranuras vuelven a estar cubiertas . Sin embargo, puede que desee que el subconjunto del clúster que funciona siga aceptando consultas de la parte del espacio de clave que sigue cubierta. Para ello, solo tiene que establecer la opción `cluster-require-full-coverage` en no.

Nombre	Detalles	Descripción
`hll-sparse-max-bytes`	Predeterminado: 3000 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Límite de bytes de representación dispersa de HyperLogLog. El límite incluye el encabezado de 16 bytes. Cuando un HyperLogLog que usa representación dispersa supera este límite, se convierte a representación densa. No se recomienda usar un valor superior a 16 000, ya que en ese punto, la representación densa es más eficaz desde el punto de vista de la memoria. Se recomienda usar un valor próximo a 3000 con el fin de disponer de los beneficios de la codificación eficaz desde el punto de vista del espacio sin ralentizar PFADD, que está habilitado con la codificación dispersa. El valor puede incrementarse hasta aproximadamente 10 000 cuando la CPU no sea un problema, pero sí lo sea el espacio y cuando el conjunto de datos esté compuesto por numerosos HyperLogLogs con la cardinalidad entre 0 y 15 000.

Nombre	Detalles	Descripción
`reserved-memory-percent`	Predeterminado: 25 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Porcentaje de memoria reservada para un nodo y no destinado a los datos. De forma predeterminada, el volumen de datos que consume Redis OSS aumenta hasta consumir toda la memoria del nodo. Si esto ocurre, es posible que el desempeño del nodo se vea afectado debido a un exceso de paginación de la memoria. Al reservar memoria, puede guardar parte de la memoria disponible para fines no relacionados con Redis OSS. De este modo, ayudará a reducir el volumen de paginación. Este parámetro es específico de ElastiCache y no forma parte de la distribución estándar de Redis OSS. Para obtener más información, consulte `reserved-memory` y Administración de la memoria reservada para Valkey y Redis OSS.

Parámetros cambiados en Redis OSS 3.2.4 (mejorado)

Familia del grupo de parámetros: redis3.2

En Redis OSS 3.2.4 se han cambiado los siguientes parámetros.

Nombre	Detalles	Cambio
`activerehashing`	Modificable: sí, si el grupo de parámetros no está asociado con ningún clúster. De lo contrario, no lo es.	Modificable era No.

Nombre	Detalles	Cambio
`databases`	Modificable: sí, si el grupo de parámetros no está asociado con ningún clúster. De lo contrario, no lo es.	Modificable era No.
`appendonly`	Valor predeterminado: desactivado Modificable: no	Si desea actualizar desde una versión anterior de Redis OSS, primero debe desactivar `appendonly` .
`appendfsync`	Valor predeterminado: desactivado Modificable: no	Si desea actualizar desde una versión anterior de Redis OSS, primero debe desactivar `appendfsync` .
`repl-timeout`	Predeterminado: 60 Modificable: no	Ahora, este parámetro no puede modificarse y tiene un valor predeterminado de 60.
`tcp-keepalive`	Valor predeterminado: 300	El valor predeterminado anterior era 0.
`list-max-ziplist-entries`		El parámetro ya no está disponible.
`list-max-ziplist-value`		El parámetro ya no está disponible.

Parámetros que se han añadido a Redis OSS 2.8.24 (mejorado)

Familia del grupo de parámetros: redis2.8

En Redis OSS 2.8.24 no se admiten parámetros adicionales.

Parámetros que se han añadido a Redis OSS 2.8.23 (mejorado)

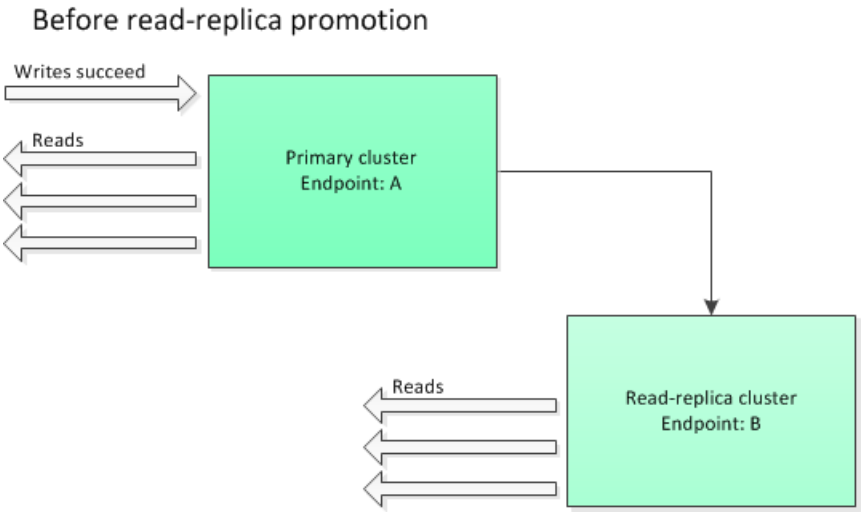
Familia del grupo de parámetros: redis2.8

En Redis OSS 2.8.23 se admite el siguiente parámetro adicional.

Nombre	Detalles	Descripción
`close-on-slave-write`	Valor predeterminado: yes Tipo: cadena (yes/no) Modificable: sí Aplicación de los cambios: inmediatamente	Si está habilitado, los clientes que intenten escribir en una réplica de solo lectura se desconectarán.

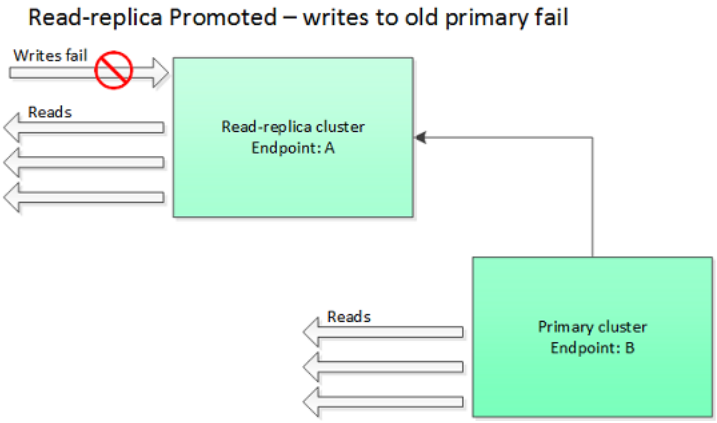
Cómo funciona close-on-slave-write

Amazon ElastiCache presenta el parámetro `close-on-slave-write` para ofrecerle más control sobre cómo responde el clúster cuando un nodo primario y un nodo de réplica de lectura cambian de rol debido a la promoción de una réplica de lectura a nodo principal.

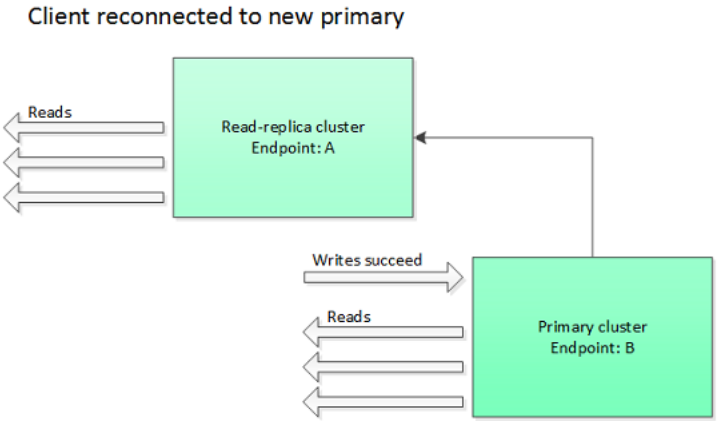


Si el clúster de réplica de lectura se promociona a nodo principal por cualquier motivo que no sea la conmutación por error de un grupo de reproducción con Multi-AZ habilitadas, el cliente seguirá intentando escribir en el punto de conexión A. Puesto que el punto de conexión A es ahora el punto de conexión de la réplica de lectura, se producirá un error en estas operaciones de escritura. Este

es el comportamiento de Redis OSS anterior a la introducción de `close-on-replica-write` en ElastiCache y el comportamiento que se obtiene al deshabilitar `close-on-replica-write`.



Con el parámetro `close-on-replica-write` habilitado, siempre que un cliente intente escribir en una réplica de lectura, se cerrará la conexión del cliente al clúster. La lógica de su aplicación debe detectar la desconexión, comprobar la tabla de DNS y volver a conectarse al punto de conexión principal, que ahora debería ser el punto de conexión B.



Cuándo debe deshabilitar close-on-replica-write

Si al deshabilitar el parámetro `close-on-replica-write` se producen errores en las operaciones de escritura en el clúster, ¿por qué deshabilitar el parámetro `close-on-replica-write`?

Tal como se ha mencionado anteriormente, con el parámetro `close-on-replica-write` habilitado, siempre que un cliente intenta escribir en una réplica de lectura, se cierra la conexión del cliente al clúster. Establecer una nueva conexión al nodo implica tiempo. Por lo tanto, la desconexión y la reconexión como resultado de una solicitud de escritura en la réplica también afecta la latencia de las solicitudes de lectura que se sirven a través de la misma conexión. Este efecto continúa hasta que se establezca una nueva conexión. Si su aplicación realiza un uso intensivo de operaciones de

lectura o es muy sensible a la latencia, es posible que desee mantener sus clientes conectados para que el rendimiento de lectura no se vea afectado.

Parámetros que se han añadido a Redis OSS 2.8.22 (mejorado)

Familia del grupo de parámetros: redis2.8

En Redis OSS 2.8.22 no se admiten parámetros adicionales.

> **Important**
>
> - A partir de Redis OSS 2.8.22, `repl-backlog-size` se aplica al clúster principal, así como a los clústeres de réplica.
> - A partir de Redis OSS 2.8.22, el parámetro `repl-timeout` no es compatible. Si se modifica, ElastiCache lo sobrescribirá con el valor predeterminado (60 s), al igual que sucede con `appendonly`.

A continuación se describen parámetros que ya no son compatibles.

- appendonly
- appendfsync
- repl-timeout

Parámetros que se han añadido a Redis OSS 2.8.21

Familia del grupo de parámetros: redis2.8

En Redis OSS 2.8.21 no se admiten parámetros adicionales.

Parámetros que se han añadido a Redis OSS 2.8.19

Familia del grupo de parámetros: redis2.8

En Redis OSS 2.8.19 no se admiten parámetros adicionales.

Parámetros que se han añadido a Redis OSS 2.8.6

Familia del grupo de parámetros: redis2.8

En Redis OSS 2.8.6 se admiten los siguientes parámetros adicionales.

Nombre	Detalles	Descripción
`min-slaves-max-lag`	Valor predeterminado: 10 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediata mente	Número de segundos en los que el nodo principal debe recibir una solicitud de ping desde una réplica de lectura. Si este periodo de tiempo transcurre y el nodo principal no recibe ningún ping, la réplica deja de considerarse como disponible. Si el número de réplicas disponibles cae por debajo del valor min-slaves-to-write, el nodo principal dejará de aceptar operaciones de escritura. Si este parámetro o min-slave s-to-write son iguales a 0, el nodo principal siempre aceptará solicitudes de escritura, incluso cuando no haya réplicas disponibl es.
`min-slaves-to-write`	Predeterminado: 0 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediata mente	Número mínimo de réplicas de lectura que deben estar disponibl es para el nodo principal acepte operaciones de escritura de los clientes. Si el número de réplicas disponibles cae por debajo de este número, el nodo principal dejará de aceptar solicitudes de escritura. Si este parámetro o min-slave s-max-lag son iguales a 0, el nodo principal siempre aceptará solicitudes de escritura, incluso cuando no haya réplicas disponibl es.

Nombre	Detalles	Descripción
`notify-keyspace-events`	Valor predeterminado: (cadena vacía) Tipo: String Modificable: sí Aplicación de los cambios: inmediata mente	Tipos de eventos de espacio de claves de los que Redis OSS puede notificar a los clientes. Cada tipo de evento se representa con una única letra: • K: eventos de espacio de claves, publicados con un prefijo __keyspace@<db>__ • E: eventos de eventos de claves, publicados con un prefijo __keyevent@<db>__ • g: comandos genéricos no específicos, como DEL, EXPIRE, RENAME, etc. • $ — comandos de cadena • l: comandos de lista • s: comandos de establecimiento • h: comandos de hash • z: comandos de conjuntos ordenados • x: eventos caducados (eventos que se generan cada vez que una clave caduca) • e: eventos expulsados (eventos que se generan cuando

Nombre	Detalles	Descripción
		se expulsa una clave para maxmemory) • A: alias para g$lshzxe Puede usar cualquier combinación de estos tipos de eventos. Por ejemplo, AKE significa que Redis OSS puede publicar notificaciones de todos los tipos de eventos. No use caracteres distintos de los mencionados anteriormente. Si intenta hacerlo, se mostrarán mensajes de error. De forma predeterminada, este parámetro está definido como una cadena vacía, lo que significa que la notificación de eventos de espacio de claves está deshabilitada.

Nombre	Detalles	Descripción
`repl-backlog-size`	Predeterminado: 1048576 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediata mente	Tamaño, en bytes, del búfer de tareas pendientes del nodo principal. Las tareas pendientes se usan para registrar actualizaciones de los datos en el nodo principal. Cuando una réplica de lectura se conecta al nodo principal, intenta realizar una sincronización parcial (`psync`), donde se aplican los datos de las tareas pendientes para ponerse al día con el nodo principal. Si se produce un error en `psync`, es necesario realizar una sincronización completa. El valor mínimo para este parámetro es 16384. **Note** A partir de Redis OSS 2.8.22, este parámetro se aplica al clúster principal, así como a las réplicas de lectura.

Nombre	Detalles	Descripción
`repl-backlog-ttl`	Predeterminado: 3600 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediata mente	Número de segundos que el nodo principal conservará el búfer de tareas pendientes. A partir del momento en que se desconect a el último nodo de réplica, los datos de las tareas pendiente s permanecerán intactos hasta que caduque `repl-backlog-ttl`. Si la réplica no conecta al nodo principal antes de que transcurra este periodo, el nodo principal liberará el búfer de tareas pendientes. Cuando la réplica vuelva a conectarse, tendrá que realizar una sincronización completa con el nodo principal. Si este parámetro se establece en 0, el búfer de tareas pendientes no se liberará nunca.

Nombre	Detalles	Descripción
`repl-timeout`	Predeterminado: 60 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediata mente	Representa el periodo de tiempo de espera, en segundos, para las operaciones siguientes: • Transferencia de datos masiva durante la sincronización, desde la perspectiva de la réplica de lectura • Tiempo de espera del nodo principal desde la perspectiva de la réplica • Tiempo de espera de la réplica desde la perspectiva del nodo principal

Parámetros de Redis OSS 2.6.13

Familia del grupo de parámetros: redis2.6

Redis OSS 2.6.13 fue la primera versión de Redis OSS compatible con ElastiCache. La tabla siguiente muestra los parámetros de Redis OSS 2.6.13 que admite ElastiCache.

Nombre	Detalles	Descripción
`activereh ashing`	Valor predeterminado: yes Tipo: cadena (yes/no) Modificable: sí Aplicación de los cambios: en el momento de la creación	Determina si desea habilitar la característica de recombinación activa de Redis. La tabla de hash principal se recombina diez veces por segundo; cada operación de recombina ción consume 1 milisegundo de tiempo de procesamiento de la CPU. Este valor se establece al crear el grupo de parámetros. Cuando se asigne un nuevo grupo

Nombre	Detalles	Descripción
		de parámetros a un clúster, este valor debe ser el mismo tanto en el nuevo grupo de parámetro s como en el anterior.
`appendonly`	Valor predeterminado: no Tipo: String Modificable: sí Aplicación de los cambios: inmediatamente	Habilita o deshabilita la característica de archivos solo anexados (AOF) de Redis. La característica AOF captura los comandos de Redis OSS que cambian datos en la caché y se usa para recuperar determinados errores de nodo. El valor predeterminado es no, lo que significa AOF está desactivado. Establezca este parámetro en yes para habilitar AOF. Para obtener más información, consulte Mitigación de errores. **Note** La característica de archivos solo anexados (AOF) no es compatible con los nodos cache.t1.micro y cache.t2.*. Para los nodos de este tipo, se ignora el valor del parámetro `appendonly` . **Note** AOF no se permite en grupos de reproducción con Multi-AZ.

Nombre	Detalles	Descripción
`appendfsync`	Valor predeterminado: everysec Tipo: String Modificable: sí Aplicación de los cambios: inmediatamente	Cuando `appendonly` está establecido en "yes", controla la frecuencia con la que el búfer de salida de AOF se escribe en el disco: • no: el búfer vacía en el disco en función de las necesidades. • everysec: el búfer se vacía una vez por segundo. Esta es la opción predeterminada. • always: el búfer se vacía siempre que se modifican datos en el clúster. • Appendfsync no es compatible con las versiones 2.8.22 y posteriores.
`client-output-buffer-limit-normal-hard-limit`	Predeterminado: 0 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Si el búfer de salida de un cliente alcanza el número de bytes especificado, el cliente se desconectará. El valor predeterminado es cero (sin límite flexible).
`client-output-buffer-limit-normal-soft-limit`	Predeterminado: 0 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Si el búfer de salida de un cliente alcanza el número de bytes especificado, el cliente se desconectará solo si esta condición se mantiene durante `client-output-buffer-limit-normal-soft-seconds`. El valor predeterminado es cero (sin límite duro).

Nombre	Detalles	Descripción
`client-output-buffer-limit-normal-soft-seconds`	Predeterminado: 0 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Si el búfer de salida de un cliente permanece en `client-output-buffer-limit-normal-soft-limit` bytes por un periodo superior a este número de segundos, el cliente se desconectará. El valor predeterminado es cero (sin límite de tiempo).
`client-output-buffer-limit-pubsub-hard-limit`	Predeterminado: 33554432 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Para clientes de publicación/suscripción de Redis OSS: si el búfer de salida de un cliente alcanza el número de bytes especificado, el cliente se desconectará.
`client-output-buffer-limit-pubsub-soft-limit`	Predeterminado: 8388608 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Para clientes de publicación/suscripción de Redis OSS: si el búfer de salida de un cliente alcanza el número de bytes especificado, el cliente se desconectará solo si esta condición se mantiene durante `client-output-buffer-limit-pubsub-soft-seconds` .
`client-output-buffer-limit-pubsub-soft-seconds`	Predeterminado: 60 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Para clientes de publicación/suscripción de Redis OSS: si el búfer de salida de un cliente permanece en `client-output-buffer-limit-pubsub-soft-limit` bytes por un tiempo superior a este número de segundos, el cliente se desconectará.

Nombre	Detalles	Descripción
`client-output-buffer-limit-slave-hard-limit`	Valor predeterminado: para obtener información acerca de los valores, consulte Parámetros específicos de tipos de nodos de Redis OSS Tipo: número entero Modificable: no	Para réplicas de lectura de Redis OSS: si el búfer de salida de un cliente alcanza el número de bytes especificado, el cliente se desconectará.
`client-output-buffer-limit-slave-soft-limit`	Valor predeterminado: para obtener información acerca de los valores, consulte Parámetros específicos de tipos de nodos de Redis OSS Tipo: número entero Modificable: no	Para réplicas de lectura de Redis OSS: si el búfer de salida de un cliente alcanza el número de bytes especificado, el cliente se desconectará solo si esta condición se mantiene durante `client-output-buffer-limit-slave-soft-seconds` .
`client-output-buffer-limit-slave-soft-seconds`	Predeterminado: 60 Tipo: número entero Modificable: no	Para réplicas de lectura de Redis OSS: si el búfer de salida de un cliente permanece en `client-output-buffer-limit-slave-soft-limit` bytes durante un tiempo superior a esta cantidad de segundos, el cliente se desconectará.
`databases`	Predeterminado: 16 Tipo: número entero Modificable: no Aplicación de los cambios: en el momento de la creación	Número de particiones lógicas en que se dividen las bases de datos. Se recomienda mantener este valor bajo. Este valor se establece al crear el grupo de parámetros. Cuando se asigne un nuevo grupo de parámetros a un clúster, este valor debe ser el mismo tanto en el nuevo grupo de parámetros como en el anterior.

Nombre	Detalles	Descripción
`hash-max-ziplist-entries`	Predeterminado: 512 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Determina la cantidad de memoria que usan los hash. Los hash con un número de entradas inferior al especificado se almacenan con una codificación especial que permite ahorrar espacio.
`hash-max-ziplist-value`	Predeterminado: 64 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Determina la cantidad de memoria que usan los hash. Los hash con entradas de tamaño inferior al número de bytes especificado se almacenan con una codificación especial que permite ahorrar espacio.
`list-max-ziplist-entries`	Predeterminado: 512 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Determina la cantidad de memoria que se usa para las listas. Las listas con un número de entradas inferior al especificado se almacenan con una codificación especial que permite ahorrar espacio.
`list-max-ziplist-value`	Predeterminado: 64 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Determina la cantidad de memoria que se usa para las listas. Las listas con entradas de tamaño inferior al número de bytes especific ado se almacenan con una codificación especial que permite ahorrar espacio.

Nombre	Detalles	Descripción
`lua-time-limit`	Predeterminado: 5000 Tipo: número entero Modificable: no	Tiempo de ejecución máximo para un script Lua, en milisegundos, antes de que ElastiCache emprenda acciones para detener el script. Si se supera el valor de `lua-time-limit` , todos los comandos de Redis OSS devolverán un error con el formato ____-BUSY. Dado que este estado puede interferir con muchas operaciones esenciales de Redis OSS, ElastiCache emitirá primero el comando SCRIPT KILL. Si esta operación no se realiza correctamente, ElastiCache forzará el reinicio de Redis OSS.
`maxclients` Este valor se aplica a todos los tipos de instancias, excepto a los especificados.	Predeterminado: 65000 Tipo: número entero Modificable: no	Número máximo de clientes que pueden conectarse a la vez.
	t2.medium Valor predeterminado: 20000 Tipo: número entero Modificable: no	
	t2.small Valor predeterminado: 20000 Tipo: número entero Modificable: no	

Nombre	Detalles	Descripción
	t2.micro Valor predeterminado: 20000 Tipo: número entero Modificable: no	
	t4g.micro Valor predeterminado: 20000 Tipo: número entero Modificable: no	
	t3.medium Valor predeterminado: 46000 Tipo: número entero Modificable: no	
	t3.small Valor predeterminado: 46000 Tipo: número entero Modificable: no	
	t3.micro Valor predeterminado: 20000 Tipo: número entero Modificable: no	

Nombre	Detalles	Descripción
`maxmemory-policy`	Valor predeterminado: volatile-lru Tipo: String Modificable: sí Aplicación de los cambios: inmediatamente	Política de expulsión de claves cuando se alcanza el uso máximo de la memoria. Los valores válidos son: `volatile-lru | allkeys-lru | volatile-random | allkeys-random | volatile-ttl | noeviction` Para obtener más información, consulte Using Valkey or Redis OSS as an LRU cache.
`maxmemory-samples`	Valor predeterminado: 3 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Para los cálculos de elementos menos usados recientemente (LRU) y de tiempo de vida (TTL), este parámetro representa el tamaño de muestra de las claves que se deben comprobar . De forma predeterminada, Redis OSS elige 3 claves y utiliza la que se usó menos recientemente.

Nombre	Detalles	Descripción
`reserved-memory`	Predeterminado: 0 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Memoria total, en bytes, reservado para elementos que no son datos. De forma predeterminada, el tamaño del nodo de Redis OSS aumentará hasta consumir la `maxmemory` del nodo (consulte Parámetros específicos de tipos de nodos de Redis OSS). Si esto ocurre, es posible que el desempeño del nodo se vea afectado debido a un exceso de paginación de la memoria. Al reservar memoria, puede guardar parte de la memoria disponible para fines no relacionados con Redis OSS. De este modo, ayudará a reducir el volumen de paginación. Este parámetro es específico de ElastiCache y no forma parte de la distribución estándar de Redis OSS. Para obtener más información, consulte `reserved-memory-percent` y Administración de la memoria reservada para Valkey y Redis OSS.
`set-max-intset-entries`	Predeterminado: 512 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Determina la cantidad de memoria que se usa para determinados tipos de conjuntos (cadenas que son enteros en base 10 en el rango de enteros con signo de 64 bits). Estos conjuntos con un número de entradas inferior al especificado se almacenan con una codificación especial que permite ahorrar espacio.

Nombre	Detalles	Descripción
`slave-allow-chaining`	Valor predeterminado: no Tipo: String Modificable: no	Determina si una réplica de lectura de Redis OSS puede tener réplicas de lectura propias.
`slowlog-log-slower-than`	Predeterminado: 10000 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Tiempo de ejecución máximo, en microsegundos, hasta que los comandos se registren con la característica Registro lento de Redis OSS.
`slowlog-max-len`	Valor predeterminado: 128 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Tamaño máximo de la característica Registro lento de Redis OSS.
`tcp-keepalive`	Predeterminado: 0 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Si se establecen un valor distinto de cero (N), los clientes de nodo se sondearán cada N segundos para asegurarse de que siguen conectados. Con el valor predeterminado 0, el sondeo se desactiva. **Important** Algunos aspectos de este parámetro han cambiado en Redis OSS 3.2.4. Consulte Parámetros cambiados en Redis OSS 3.2.4 (mejorado).

Nombre	Detalles	Descripción
`timeout`	Predeterminado: 0 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Número de segundos que un nodo espera antes de caducar. Valores son los siguientes: • `0`: no desconectar nunca un cliente inactivo. • `1-19`: valores no válidos. • `>=20`: número de segundos que un nodo espera antes de desconectar un cliente inactivo.
`zset-max-ziplist-entries`	Valor predeterminado: 128 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Determina la cantidad de memoria que se usa para los conjuntos ordenados. Los conjuntos ordenados con un número de elementos inferior al especificado se almacenan con una codificación especial que permite ahorrar espacio.
`zset-max-ziplist-value`	Predeterminado: 64 Tipo: número entero Modificable: sí Aplicación de los cambios: inmediatamente	Determina la cantidad de memoria que se usa para los conjuntos ordenados. Los conjuntos ordenados con entradas de tamaño inferior al número de bytes especificado se almacenan con una codificación especial que permite ahorrar espacio.

Note

Si no especifica ningún grupo de parámetros para su clúster de Redis OSS 2.6.13, se usará un grupo de parámetros predeterminado (`default.redis2.6`). No puede cambiar los valores de ninguno de los parámetros del grupo de parámetros predeterminado; sin embargo, siempre podrá crear un grupo de parámetros predeterminado y asignarlo a su clúster en cualquier momento.

Parámetros específicos de tipos de nodos de Redis OSS

Aunque la mayoría de los parámetros tienen un único valor, algunos parámetros tienen distintos valores en función del tipo de nodo que se use. La tabla siguiente muestra los valores predeterminados de los parámetros `maxmemory`, `client-output-buffer-limit-slave-hard-limit` y `client-output-buffer-limit-slave-soft-limit` para cada tipo de nodo. El valor de `maxmemory` es el número máximo de bytes disponibles para el uso, los datos y otros usos en el nodo. Para obtener más información, consulte Memoria disponible.

> **Note**
>
> El parámetro `maxmemory` no se puede modificar.

Tipo de nodo	Maxmemory	Client-output-buffer-limit-slave-hard-limit	Client-output-buffer-limit-slave-soft-limit
cache.t1.micro	142606336	14260633	14260633
cache.t2.micro	581959680	58195968	58195968
cache.t2.small	1665138688	166513868	166513868
cache.t2.medium	3461349376	346134937	346134937
cache.t3.micro	536870912	53687091	53687091
cache.t3.small	1471026299	147102629	147102629
cache.t3.medium	3317862236	331786223	331786223
cache.t4g.micro	536870912	53687091	53687091
cache.t4g.small	1471026299	147102629	147102629
cache.t4g.medium	3317862236	331786223	331786223
cache.m1.small	943718400	94371840	94371840

Tipo de nodo	Maxmemory	Client-output-buffer-limit-slave-hard-limit	Client-output-buffer-limit-slave-soft-limit
cache.m1.medium	3093299200	309329920	309329920
cache.m1.large	7025459200	702545920	702545920
cache.m1.xlarge	14889779200	1488977920	1488977920
cache.m2.xlarge	17091788800	1709178880	1709178880
cache.m2.2xlarge	35022438400	3502243840	3502243840
cache.m2.4xlarge	70883737600	7088373760	7088373760
cache.m3.medium	2988441600	309329920	309329920
cache.m3.large	6501171200	650117120	650117120
cache.m3.xlarge	14260633600	1426063360	1426063360
cache.m3.2xlarge	29989273600	2998927360	2998927360
cache.m4.large	6892593152	689259315	689259315
cache.m4.xlarge	15328501760	1532850176	1532850176
cache.m4.2xlarge	31889126359	3188912636	3188912636
cache.m4.4xlarge	65257290629	6525729063	6525729063
cache.m4.10xlarge	166047614239	16604761424	16604761424
cache.m5.large	6854542746	685454275	685454275
cache.m5.xlarge	13891921715	1389192172	1389192172
cache.m5.2xlarge	27966669210	2796666921	2796666921
cache.m5.4xlarge	56116178125	5611617812	5611617812

Tipo de nodo	Maxmemory	Client-output-buffer-limit-slave-hard-limit	Client-output-buffer-limit-slave-soft-limit
cache.m5.12xlarge	168715971994	16871597199	16871597199
cache.m5.24xlarge	337500562842	33750056284	33750056284
cache.m6g.large	6854542746	685454275	685454275
cache.m6g.xlarge	13891921715	1389192172	1389192172
cache.m6g.2xlarge	27966669210	2796666921	2796666921
cache.m6g.4xlarge	56116178125	5611617812	5611617812
cache.m6g.8xlarge	111325552312	11132555231	11132555231
cache.m6g.12xlarge	168715971994	16871597199	16871597199
cache.m6g.16xlarge	225000375228	22500037523	22500037523
cache.c1.xlarge	6501171200	650117120	650117120
cache.r3.large	14470348800	1468006400	1468006400
cache.r3.xlarge	30513561600	3040870400	3040870400
cache.r3.2xlarge	62495129600	6081740800	6081740800
cache.r3.4xlarge	126458265600	12268339200	12268339200
cache.r3.8xlarge	254384537600	24536678400	24536678400
cache.r4.large	13201781556	1320178155	1320178155
cache.r4.xlarge	26898228839	2689822883	2689822883
cache.r4.2xlarge	54197537997	5419753799	5419753799
cache.r4.4xlarge	108858546586	10885854658	10885854658

Tipo de nodo	Maxmemory	Client-output-buffer-limit-slave-hard-limit	Client-output-buffer-limit-slave-soft-limit
cache.r4.8xlarge	218255432090	21825543209	21825543209
cache.r4.16xlarge	437021573120	43702157312	43702157312
cache.r5.large	14037181030	1403718103	1403718103
cache.r5.xlarge	28261849702	2826184970	2826184970
cache.r5.2xlarge	56711183565	5671118356	5671118356
cache.r5.4xlarge	113609865216	11360986522	11360986522
cache.r5.12xlarge	341206346547	34120634655	34120634655
cache.r5.24xlarge	682485973811	68248597381	68248597381
cache.r6g.large	14037181030	1403718103	1403718103
cache.r6g.xlarge	28261849702	2826184970	2826184970
cache.r6g.2xlarge	56711183565	5671118356	5671118356
cache.r6g.4xlarge	113609865216	11360986522	11360986522
cache.r6g.8xlarge	225000375228	22500037523	22500037523
cache.r6g.12xlarge	341206346547	34120634655	34120634655
cache.r6g.16xlarge	450000750456	45000075046	45000075046
cache.r6gd.xlarge	28261849702	2826184970	2826184970
cache.r6gd.2xlarge	56711183565	5671118356	5671118356
cache.r6gd.4xlarge	113609865216	11360986522	11360986522
cache.r6gd.8xlarge	225000375228	22500037523	22500037523

Tipo de nodo	Maxmemory	Client-output-buffer-limit-slave-hard-limit	Client-output-buffer-limit-slave-soft-limit
cache.r6gd.12xlarge	341206346547	34120634655	34120634655
cache.r6gd.16xlarge	450000750456	45000075046	45000075046
cache.r7g.large	14037181030	1403718103	1403718103
cache.r7g.xlarge	28261849702	2826184970	2826184970
cache.r7g.2xlarge	56711183565	5671118356	5671118356
cache.r7g.4xlarge	113609865216	11360986522	11360986522
cache.r7g.8xlarge	225000375228	22500037523	22500037523
cache.r7g.12xlarge	341206346547	34120634655	34120634655
cache.r7g.16xlarge	450000750456	45000075046	45000075046
cache.m7g.large	6854542746	685454275	685454275
cache.m7g.xlarge	13891921715	1389192172	1389192172
cache.m7g.2xlarge	27966669210	2796666921	2796666921
cache.m7g.4xlarge	56116178125	5611617812	5611617812
cache.m7g.8xlarge	111325552312	11132555231	11132555231
cache.m7g.12xlarge	168715971994	16871597199	16871597199
cache.m7g.16xlarge	225000375228	22500037523	22500037523
cache.c7gn.large	3317862236	1403718103	1403718103
cache.c7gn.xlarge	6854542746	2826184970	2826184970
cache.c7gn.2xlarge	13891921715	5671118356	5671118356

Tipo de nodo	Maxmemory	Client-output-buffer-limit-slave-hard-limit	Client-output-buffer-limit-slave-soft-limit
cache.c7gn.4xlarge	27966669210	11360986522	11360986522
cache.c7gn.8xlarge	56116178125	22500037523	22500037523
cache.c7gn.12xlarge	84357985997	34120634655	34120634655
cache.c7gn.16xlarge	113609865216	45000075046	45000075046

> **Note**
>
> Todos los tipos de instancias de la generación actual se crean en Amazon Virtual Private Cloud (VPC) de forma predeterminada.
> Las instancias T1 no admiten Multi-AZ.
> Las instancias T1 y T2 no admiten AOF de Redis OSS.
> Las variables de configuración de Redis OSS `appendonly` y `appendfsync` no se admiten en la versión 2.8.22 de Redis OSS y las versiones posteriores.

Parámetros específicos de Memcached

Memcached

Si no se especifica ningún grupo de parámetros para el clúster de Memcached, se usará un grupo de parámetros predeterminado apropiado para la versión del motor. No puede cambiar los valores de los parámetros de un grupo de parámetros predeterminado. Sin embargo, puede crear un grupo de parámetros personalizados y asignarlo a su clúster en cualquier momento. Para obtener más información, consulte Creación de un grupo de parámetros de ElastiCache.

Temas

- Cambios en Memcached 1.6.17
- Parámetros agregados a Memcached 1.6.6
- Cambios en los parámetros de Memcached 1.5.10
- Parámetros agregados a Memcached 1.4.34

- Parámetros agregados a Memcached 1.4.33
- Parámetros agregados a Memcached 1.4.24
- Parámetros agregados a Memcached 1.4.14
- Parámetros compatibles con Memcached 1.4.5
- Capacidad adicional para conexiones de Memcached
- Parámetros específicos de tipo de nodo de Memcached

Cambios en Memcached 1.6.17

A partir de la versión 1.6.17 de Memcached, ya no se admiten estos comandos administrativos: `lru_crawler`, `lru` y `slabs`. Con estos cambios, no podrá habilitar o deshabilitar `lru_crawler` en el tiempo de ejecución mediante comandos. Habilite o deshabilite `lru_crawler` modificando su grupo de parámetros personalizado.

Parámetros agregados a Memcached 1.6.6

Para Memcached 1.6.6, no se admiten parámetros adicionales.

Familia del grupo de parámetros: memcached1.6

Cambios en los parámetros de Memcached 1.5.10

Para Memcached 1.5.10, se admiten los siguientes parámetros adicionales.

Familia del grupo de parámetros: memcached1.5

Nombre	Detalles	Descripción
`no_modern`	Valor predeterminado: 1 Tipo: booleano Modificable: sí Valores permitidos: 0, 1	Un alias para deshabilitar los comandos `slab_reassign` , `lru_maintainer_thread` , `lru_segmented` y `maxconns_fast` . Al usar Memcached 1.5 y versiones posteriores, `no_modern`

Nombre	Detalles	Descripción
	Aplicación de los cambios: en el momento del lanzamiento	también establece hash_algorithm en `jenkins`. Además, cuando se utiliza Memcached 1.5.10, `inline_ascii_reponse` se controla mediante el parámetro `parallelly` . Esto significa que si `no_modern` está deshabilitado, `inline_ascii_response` también está deshabilitado. A partir del motor 1.5.16 de Memcached, el parámetro `inline_ascii_response` ya no se aplica, por lo que habilitar o deshabilitar `no_modern` no afecta a `inline_ascii_response` . Si `no_modern` está deshabilitado, se HABILITARÁ `slab_reassign` , `lru_maintainer_thread` , `lru_segmented` y `maxconns_fast` . Dado que los parámetros `slab_automove` y `hash_algorithm` no son parámetros de SWITCH, sus ajustes se basan en las configuraciones del grupo de parámetros. Si desea deshabilitar `no_modern` y volver a `modern`, debe configurar un grupo de parámetros personalizado para deshabilitar

Nombre	Detalles	Descripción
		este parámetro y, a continuación, reiniciarlo para que los cambios se apliquen. **Note** El valor de configuración predeterminado para este parámetro se ha cambiado de 0 a 1 a partir del 20 de agosto de 2021. Los nuevos usuarios de ElastiCache obtendrán automáticamente el valor predeterminado actualizado para cada región a partir del 20 de agosto de 2021. Los usuarios de ElastiCache existentes en las regiones antes del 20 de agosto de 2021 deben modificar manualmente sus grupos de parámetros personalizados para poder realizar este nuevo cambio.

Nombre	Detalles	Descripción
`inline_ascii_resp`	Predeterminado: 0 Tipo: booleano Modificable: sí Valores permitidos: 0, 1 Aplicación de los cambios: en el momento del lanzamiento	Almacena números de la respuesta de `VALUE`, dentro de un elemento, utilizando hasta 24 bytes. Supone una ligera ralentización para las operaciones `get` y `faster` en ASCII.

Para Memcached 1.5.10, se han eliminado los parámetros siguientes.

Nombre	Detalles	Descripción
`expirezero_does_not_evict`	Predeterminado: 0 Tipo: booleano Modificable: sí Valores permitidos: 0, 1 Aplicación de los cambios: en el momento del lanzamiento	Ya no se admite en esta versión.
`modern`	Valor predeterminado: 1 Tipo: booleano	Ya no se admite en esta versión. A partir de esta versión, `no-modern` está habilitado de

Nombre	Detalles	Descripción
	Modificable: sí (requiere volver a lanzar si se establece en `no_modern`) Valores permitidos: 0, 1 Aplicación de los cambios: en el momento del lanzamiento	forma predeterminada con cada lanzamiento o relanzamiento.

Parámetros agregados a Memcached 1.4.34

Para Memcached 1.4.34, no se admite ningún parámetro adicional.

Familia del grupo de parámetros: memcached1.4

Parámetros agregados a Memcached 1.4.33

Para Memcached 1.4.33, se admiten los siguientes parámetros adicionales.

Familia del grupo de parámetros: memcached1.4

Nombre	Detalles	Descripción
`modern`	Valor predeterminado: habilitado Tipo: booleano Modificable: sí Aplicación de los cambios: en el momento del lanzamiento	Alias para varias características. Habilitar `modern` equivale a habilitar los comandos siguientes y usar un algoritmo de hash murmur3: `slab_reassign` , `slab_automove` , `lru_crawler` , `lru_maintainer` , `maxconns_fast` y `hash_algorithm=murmur3` .

Nombre	Detalles	Descripción
`watch`	Valor predeterminado: habilitado Tipo: booleano Modificable: sí Aplicación de los cambios: inmediatamente Los logs pueden eliminarse si el usuario alcanza los límites `watcher_logbuf_size` y `worker_logbuf_size` .	Recuperaciones de logs, expulsiones o mutaciones. Cuando, por ejemplo, el usuario activa `watch`, podrá ver los logs cuando tengan lugar las operaciones `get`, `set`, `delete` o `update`.
`idle_timeout`	Valor predeterminado: 0 (deshabilitado) Tipo: número entero Modificable: sí Aplicación de los cambios: en el momento del lanzamiento	Número mínimo de segundos que se permitirá que un cliente permanezca inactivo antes de solicitar el cierre. Rango de valores: de 0 a 86400.

Nombre	Detalles	Descripción
`track_sizes`	Valor predeterminado: deshabilitado Tipo: booleano Modificable: sí Aplicación de los cambios: en el momento del lanzamiento	Muestra el tamaño que cada grupo de slab ha consumido. Habilitar `track_sizes` le permite ejecutar `stats sizes` sin necesidad de ejecutar `stats sizes_enable` .
`watcher_logbuf_size`	Valor predeterminado: 256 (KB) Tipo: número entero Modificable: sí Aplicación de los cambios: en el momento del lanzamiento	El comando `watch` activa el registro de transmisión para Memcached. Sin embargo, `watch` puede eliminar logs si la tasa de expulsiones, mutaciones o recuperaciones es lo suficientemente alta como para llenar el búfer de registro. En estas situaciones, los usuarios pueden aumentar el tamaño del búfer para reducir la posibilidad de que se produzcan pérdidas de logs.

Nombre	Detalles	Descripción
`worker_logbuf_size`	Valor predeterminado: 64 (KB) Tipo: número entero Modificable: sí Aplicación de los cambios: en el momento del lanzamiento	El comando `watch` activa el registro de transmisión para Memcached. Sin embargo, `watch` puede eliminar logs si la tasa de expulsiones, mutaciones o recuperaciones es lo suficient emente alta como para llenar el búfer de registro. En estas situaciones, los usuarios pueden aumentar el tamaño del búfer para reducir la posibilidad de que se produzcan pérdidas de logs.
`slab_chunk_max`	Valor predeterminado: 524288 (bytes) Tipo: número entero Modificable: sí Aplicación de los cambios: en el momento del lanzamiento	Especifica el tamaño de slab máximo. Establecer un tamaño de slab más reducido permite usar la memoria de un modo más eficaz. Los elementos cuyo tamaño supere `slab_chunk_max` , se dividen en múltiples slabs.
`lru_crawler metadump [all|1|2|3]`	Valor predeterminado: deshabilitado Tipo: booleano Modificable: sí Aplicación de los cambios: inmediata mente	Si lru_crawler está habilitado, este comando volcará todas las claves. `all|1|2|3` : todos los slabs; o bien especifique un número de slab específico

Parámetros agregados a Memcached 1.4.24

Para Memcached 1.4.24, se admiten los siguientes parámetros adicionales.

Familia del grupo de parámetros: memcached1.4

Nombre	Detalles	Descripción
`disable_flush_all`	Valor predeterminado: 0 (deshabilitado) Tipo: booleano Modificable: sí Aplicación de los cambios: en el momento del lanzamiento	Agregue el parámetro (`-F`) para deshabilitar flush_all. Esto resulta útil si no quiere que se pueda ejecutar un vaciado completo en instancias de producción. Valores: 0, 1 (el usuario puede usar `flush_all` cuando el valor es 0).
`hash_algorithm`	Valor predeterminado: jenkins Tipo: String Modificable: sí Aplicación de los cambios: en el momento del lanzamiento	Algoritmo de hash que se va a usar. Valores permitidos: murmur3 y jenkins.
`lru_crawler`	Valor predeterminado: 0 (deshabilitado) Tipo: booleano Modificable: sí Aplicación de los cambios: tras reiniciar	Limpia clases de slab de elementos que han caducado. Este es un proceso de bajo impacto que se ejecuta en segundo plano. En la actualida d, es necesario iniciar un rastreo mediante un comando manual.

Nombre	Detalles	Descripción
	Note Puede habilitar temporalmente `lru_crawler` durante el tiempo de ejecución desde la línea de comandos. Para obtener más información, consulte la columna Descripción.	Para habilitar el proceso temporalmente, ejecute `lru_crawler enable` en la línea de comandos. `lru_crawler 1,3,5` rastrea clases de slab 1, 3 y 5 buscando elementos caducados para agregarlos a la lista libre. Valores: 0, 1 **Note** Habilitar `lru_crawler` en la línea de comandos, se habilita el rastreador hasta que se deshabilita en la línea de comandos o hasta el siguiente reinicio. Para habilitar el proceso de forma permanente, debe modificar el valor del parámetro. Para obtener más información, consulte Modificación de un grupo de parámetros de ElastiCache.

Nombre	Detalles	Descripción
`lru_maintainer`	Valor predeterminado: 0 (deshabilitado) Tipo: booleano Modificable: sí Aplicación de los cambios: en el momento del lanzamiento	Subproceso que se ejecuta en segundo plano y que cambia los elementos entre LRU a medida que se agota la capacidad. Valores: 0, 1
`expirezero_does_not_evict`	Valor predeterminado: 0 (deshabilitado) Tipo: booleano Modificable: sí Aplicación de los cambios: en el momento del lanzamiento	Cuando se usa con `lru_maintainer` , los elementos cuyo periodo de vencimiento es 0 no se pueden expulsar. ⚠ Warning Esto puede agotar la memoria disponible para otros elementos que se pueden expulsar. Se puede establecer para ignorar `lru_maintainer` .

Parámetros agregados a Memcached 1.4.14

Para Memcached 1.4.14, se admiten los siguientes parámetros adicionales.

Familia del grupo de parámetros: memcached1.4

Parámetros agregados a Memcached 1.4.14

Nombre	Descripción
`config_max`	El número máximo de entradas de configuración de ElastiCache.
`config_size_max`	Tamaño máximo de las entradas de configuración en bytes.
`hashpower_init`	El tamaño inicial de la tabla hash de ElastiCache, expresado en potencia de dos. El valor predeterminado es 16 (2^16) o 65536 claves.

Nombre	Descripción
`maxconns_fast`	Cambia el modo en que se gestionan las solicitudes de nuevas conexiones cuando se alcanza el límite máximo de la conexión. Si este parámetro se establece en 0 (cero), las conexiones nuevas se agregan a la cola de tareas pendientes a la espera de que se cierren otras conexiones. Si el parámetro se establece en 1, ElastiCache envía un error al cliente y cierra la conexión de inmediato.

Nombre	Descripción
slab_automove	Ajusta el algoritmo automove de los slabs: si este parámetro se establece en 0 (cero), el algoritmo automove estará deshabilitado. Si se establece en 1, ElastiCache adoptará un enfoque lento y conservador para mover los bloques de forma automática. Si se establece en 2, ElastiCache mueve los bloques de forma agresiva siempre que se produce una expulsión. (Este modo no se recomienda, excepto con fines de prueba.)

Nombre	Descripción
slab_reassign	Habilita o deshabilita la reasignación de slab. Si este parámetro se establece en 1, podrá usar el comando "slabs reassign" para reasignar manualmente la memoria.

Parámetros compatibles con Memcached 1.4.5

Familia del grupo de parámetros: memcached1.4

Para Memcached 1.4.5, se admiten los siguientes parámetros.

Parámetros agregados a Memcached 1.4.5

Nombre	Detalles	Descripción
backlog_queue_limit	Predeterminado: 1024 Tipo: número entero Modificable: no	Límite de cola de tareas pendientes.

Nombre	Detalles	Descripción
`binding_protocol`	Valor predeterminado: automático Tipo: String Modificable: sí Aplicación de los cambios: tras reiniciar	Protocolo de enlace. Valores permitidos: `ascii` y `auto`. Para obtener información acerca de la modificación del valor de `binding_protocol` , consulte Modificación de un grupo de parámetros de ElastiCache.
`cas_disabled`	Valor predeterminado: 0 (falso) Tipo: booleano Modificable: sí Aplicación de los cambios: tras reiniciar	Si se establece en 1 (true), compruebe y defina (CAS) las operaciones que se deshabilitarán. Los elementos almacenados ocuparán 8 bytes menos que con CAS habilitado.
`chunk_size`	Predeterminado: 48 Tipo: número entero Modificable: sí Aplicación de los cambios: tras reiniciar	Cantidad mínima, en bytes, de espacio para asignar a las marcas, los valores y las claves de elemento más pequeños.
`chunk_size_growth_factor`	Predeterminado: 1.25 Tipo: flotante Modificable: sí Aplicación de los cambios: tras reiniciar	Factor de crecimiento que controla el tamaño de cada fragmento de Memcached sucesivo; cada fragmento tendrá un tamaño `chunk_size_growth_factor` veces mayor que el anterior.

Nombre	Detalles	Descripción
`error_on_memory_exhausted`	Valor predeterminado: 0 (falso) Tipo: booleano Modificable: sí Aplicación de los cambios: tras reiniciar	Si se establece en 1 (true), cuando no haya más memoria para almacenar elementos, Memcached devolverá un mensaje de error en lugar de expulsar elementos.
`large_memory_pages`	Valor predeterminado: 0 (falso) Tipo: booleano Modificable: no	Si se establece en 1 (true [verdadero]), ElastiCache intentará utilizar páginas de gran memoria.
`lock_down_paged_memory`	Valor predeterminado: 0 (falso) Tipo: booleano Modificable: no	Si se establece en 1 (true [verdadero]), ElastiCache bloqueará toda la memoria paginada.
`max_item_size`	Predeterminado: 1048576 Tipo: número entero Modificable: sí Aplicación de los cambios: tras reiniciar	Tamaño, en bytes, de los elementos de mayor tamaño que se pueden almacenar en el clúster.
`max_simultaneous_connections`	Predeterminado: 65000 Tipo: número entero Modificable: no	Número máximo de conexiones simultáneas.

Nombre	Detalles	Descripción
`maximize_core_file_limit`	Valor predeterminado: 0 (falso) Tipo: booleano Modificable: Aplicación de los cambios: tras reiniciar	Si se establece en 1 (true [verdadero]), ElastiCache maximizará el límite de archivo de núcleo.
`memcached_connections_overhead`	Predeterminado: 100 Tipo: número entero Modificable: sí Aplicación de los cambios: tras reiniciar	Cantidad de memoria reservada para las conexiones de Memcached y capacidad adicional para otras operaciones. Para obtener información acerca de este parámetro, consulte Capacidad adicional para conexiones de Memcached.
`requests_per_event`	Predeterminado: 20 Tipo: número entero Modificable: no	Número máximo de solicitudes por evento para una conexión determinada. Este límite es necesario para evitar el agotamiento de los recursos.

Capacidad adicional para conexiones de Memcached

En cada nodo, la memoria disponible para almacenar elementos equivale a la cantidad total de memoria disponible de dicho nodo (que se almacena en el parámetro `max_cache_memory`) menos la memoria que se usa para las conexiones y la capacidad adicional para otras operaciones (que se almacena en el parámetro `memcached_connections_overhead`). Por ejemplo, un nodo del tipo `cache.m1.small` tiene una `max_cache_memory` de 1300 MB. Con el valor `memcached_connections_overhead` predeterminado de 100 MB, el proceso de Memcached tendrá 1200 MB de espacio disponible para almacenar elementos.

Los valores predeterminados para el parámetro `memcached_connections_overhead` son válidos para la mayoría de los casos de uso; sin embargo, la asignación necesaria para la capacidad

adicional de conexión puede variar dependiendo de varios factores, incluida la tasa de solicitudes, el tamaño de carga y el número de conexiones.

Puede cambiar el valor de `memcached_connections_overhead` según las necesidades de su aplicación. Por ejemplo, al aumentar el valor del parámetro `memcached_connections_overhead`, se reducirá la cantidad de memoria disponible para almacenar elementos y dispondrá de un búfer de mayor tamaño para la sobrecarga de las conexiones. La reducción del valor del parámetro `memcached_connections_overhead` le dará más memoria para almacenar elementos, pero puede aumentar el riesgo de agotar el espacio de intercambio y de que se reduzca el rendimiento. Si se agota el espacio de intercambio y observa una pérdida de rendimiento, pruebe incrementar el valor del parámetro `memcached_connections_overhead`.

> **Important**
>
> Para el tipo nodo `cache.t1.micro`, el valor de `memcached_connections_overhead` se determina de la manera siguiente:
>
> - Si su clúster utiliza el grupo de parámetros predeterminado, ElastiCache establecerá el valor de `memcached_connections_overhead` en 13 MB.
> - Si su clúster usa un grupo de parámetros de su creación, podrá establecer el valor de `memcached_connections_overhead` según su preferencia.

Parámetros específicos de tipo de nodo de Memcached

Aunque la mayoría de los parámetros tienen un único valor, algunos parámetros tienen distintos valores en función del tipo de nodo que se use. La tabla siguiente muestra los valores predeterminados de los parámetros `max_cache_memory` y `num_threads` para cada tipo de nodo. Los valores de estos parámetros no se pueden modificar.

Tipo de nodo	max_cache_memory (en megabytes)	num_threads
cache.t1.micro	213	1
cache.t2.micro	555	1
cache.t2.small	1588	1

Tipo de nodo	max_cache_memory (en megabytes)	num_threads
cache.t2.medium	3301	2
cache.t3.micro	512	2
cache.t3.small	1402	2
cache.t3.medium	3364	2
cache.t4g.micro	512	2
cache.t4g.small	1402	2
cache.t4g.medium	3164	2
cache.m1.small	1301	1
cache.m1.medium	3350	1
cache.m1.large	7100	2
cache.m1.xlarge	14600	4
cache.m2.xlarge	33800	2
cache.m2.2xlarge	30412	4
cache.m2.4xlarge	68000	16
cache.m3.medium	2850	1
cache.m3.large	6200	2
cache.m3.xlarge	13600	4
cache.m3.2xlarge	28600	8
cache.m4.large	6573	2
cache.m4.xlarge	11496	4

Tipo de nodo	max_cache_memory (en megabytes)	num_threads
cache.m4.2xlarge	30412	8
cache.m4.4xlarge	62234	16
cache.m4.10xlarge	158355	40
cache.m5.large	6537	2
cache.m5.xlarge	13248	4
cache.m5.2xlarge	26671	8
cache.m5.4xlarge	53516	16
cache.m5.12xlarge	160900	48
cache.m5.24xlarge	321865	96
cache.m6g.large	6537	2
cache.m6g.xlarge	13248	4
cache.m6g.2xlarge	26671	8
cache.m6g.4xlarge	53516	16
cache.m6g.8xlarge	107000	32
cache.m6g.12xlarge	160900	48
cache.m6g.16xlarge	214577	64
cache.c1.xlarge	6600	8
cache.r3.large	13800	2
cache.r3.xlarge	29100	4
cache.r3.2xlarge	59600	8

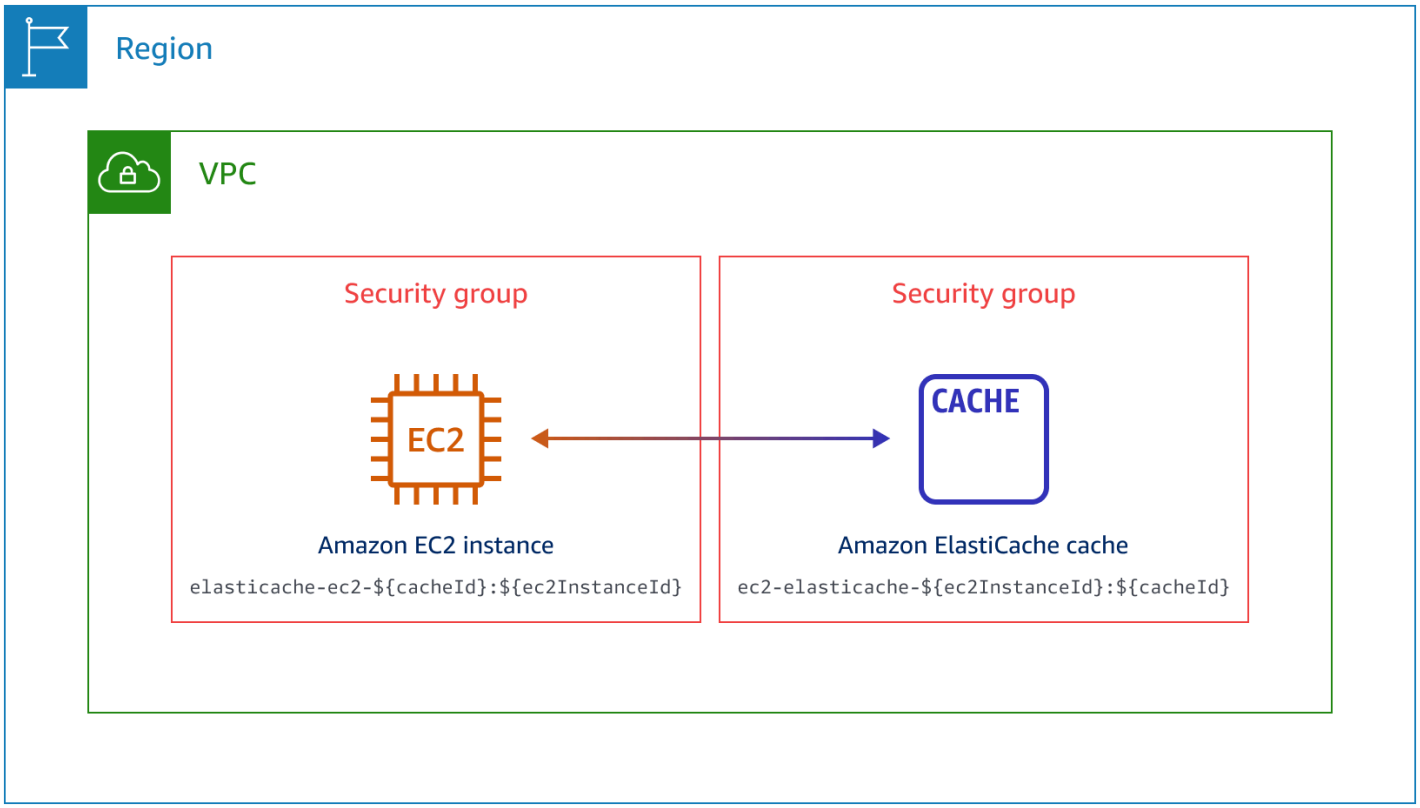
Tipo de nodo	max_cache_memory (en megabytes)	num_threads
cache.r3.4xlarge	120600	16
cache.r3.8xlarge	120600	32
cache.r4.large	12590	2
cache.r4.xlarge	25652	4
cache.r4.2xlarge	51686	8
cache.r4.4xlarge	103815	16
cache.r4.8xlarge	208144	32
cache.r4.16xlarge	416776	64
cache.r5.large	13387	2
cache.r5.xlarge	26953	4
cache.r5.2xlarge	54084	8
cache.r5.4xlarge	108347	16
cache.r5.12xlarge	325400	48
cache.r5.24xlarge	650869	96
cache.r6g.large	13387	2
cache.r6g.xlarge	26953	4
cache.r6g.2xlarge	54084	8
cache.r6g.4xlarge	108347	16
cache.r6g.8xlarge	214577	32
cache.r6g.12xlarge	325400	48

Tipo de nodo	max_cache_memory (en megabytes)	num_threads
cache.r6g.16xlarge	429154	64
cache.c7gn.large	3164	2
cache.c7gn.xlarge	6537	4
cache.c7gn.2xlarge	13248	8
cache.c7gn.4xlarge	26671	16
cache.c7gn.8xlarge	53516	32
cache.c7gn.12xlarge	325400	48
cache.c7gn.16xlarge	108347	64

> **Note**
>
> Todas las instancias T2 se crean en una instancia de Amazon Virtual Private Cloud (Amazon VPC).

Conexión automática de una instancia de EC2 y una caché de ElastiCache

Puede utilizar la consola de ElastiCache para simplificar la configuración de una conexión entre una instancia de Amazon Elastic Compute Cloud (Amazon EC2) y una caché de ElastiCache. A menudo, la caché se encuentra en una subred privada y la instancia de EC2 en una subred pública dentro de una VPC. Puede usar un cliente SQL en su instancia de EC2 para conectarse a la caché de ElastiCache. La instancia de EC2 también puede ejecutar servidores web o aplicaciones que accedan a la caché privada de ElastiCache.



Temas

- Conectividad automática con una instancia de EC2
- Visualización de los recursos de computación conectados

Conectividad automática con una instancia de EC2

Cuando se configura una conexión entre una instancia de EC2 y una caché de ElastiCache, ElastiCache configura automáticamente el grupo de seguridad de la VPC para la instancia de EC2 y la caché de ElastiCache.

Estos son los requisitos para conectar una instancia de EC2 con una caché de ElastiCache:

- La instancia de EC2 debe existir en la misma VPC que la caché de ElastiCache.

  Si no existen instancias de EC2 en la misma VPC, la consola proporciona un enlace para crear una.
- El usuario que establece la conectividad debe tener permisos para realizar las siguientes operaciones de Amazon EC2. Por lo general, estos permisos se agregan a las cuentas de

EC2 cuando se crean. Para obtener más información acerca de los permisos de EC2, consulte Concesión de los permisos necesarios para los recursos de Amazon EC2.

- `ec2:AuthorizeSecurityGroupEgress`
- `ec2:AuthorizeSecurityGroupIngress`
- `ec2:CreateSecurityGroup`
- `ec2:DescribeInstances`
- `ec2:DescribeNetworkInterfaces`
- `ec2:DescribeSecurityGroups`
- `ec2:ModifyNetworkInterfaceAttribute`
- `ec2:RevokeSecurityGroupEgress`

Cuando se establece una conexión con una instancia de EC2, ElastiCache realiza una acción basada en la configuración actual de los grupos de seguridad asociados a la caché de ElastiCache y la instancia de EC2, tal como se describe en la siguiente tabla.

Configuración actual del grupo de seguridad de ElastiCache	Configuración del grupo de seguridad de EC2 actual	Acción de ElastiCache
Hay uno o más grupos de seguridad asociados a la caché de ElastiCache con un nombre que coincide con el patrón `elasticache-ec2-${cacheId}:${ec2InstanceId}` . No se ha modificado ningún grupo de seguridad que coincida con el patrón. Este grupo de seguridad tiene solo una regla de entrada con el grupo de seguridad de VPC de la instancia de EC2 como origen.	Hay uno o más grupos de seguridad asociados a la instancia de EC2 con un nombre que coincide con el patrón `elasticache-ec2-${cacheId}:${ec2InstanceId}` . No se ha modificado ningún grupo de seguridad que coincida con el patrón. Este grupo de seguridad solo tiene una regla de salida con el grupo de seguridad de la VPC de la caché de ElastiCache como origen.	ElastiCache no realiza ninguna acción. Ya se configuró automáticamente una conexión entre la instancia de EC2 y la caché de ElastiCache. Como ya existe una conexión entre la instancia de EC2 y la caché de ElastiCache, los grupos de seguridad no se modifican.

Configuración actual del grupo de seguridad de ElastiCache	Configuración del grupo de seguridad de EC2 actual	Acción de ElastiCache
Se aplica alguna de las siguientes condiciones: • No hay ningún grupo de seguridad asociado a la caché de ElastiCache con un nombre que coincida con el patrón `elasticache-ec2-${cacheId}:${ec2InstanceId}` . • Hay uno o más grupos de seguridad asociados a la caché de ElastiCache con un nombre que coincide con el patrón `elasticache-ec2-${cacheId}:${ec2InstanceId}` . Sin embargo, ElastiCache no puede usar ninguno de estos grupos de seguridad para la conexión con la instancia de EC2. ElastiCache no puede usar un grupo de seguridad que no tenga una regla de entrada con el grupo de seguridad de la VPC de la instancia de EC2 como origen. ElastiCache tampoco puede usar un grupo de seguridad que se haya modificado. Los ejemplos de modificaciones incluyen agregar una regla	Se aplica alguna de las siguientes condiciones: • No hay ningún grupo de seguridad asociado a la instancia de EC2 con un nombre que coincida con el patrón `ec2-elasticache-${ec2InstanceId}:${cacheId}` . • Hay uno o más grupos de seguridad asociados a la instancia de EC2 con un nombre que coincide con el patrón `ec2-elasticache-${ec2InstanceId}:${cacheId}` . Sin embargo, ElastiCache no puede usar ninguno de estos grupos de seguridad para la conexión con la caché de ElastiCache. ElastiCache no puede usar un grupo de seguridad que no tenga una regla de entrada con el grupo de seguridad de la VPC de la caché de ElastiCache como origen. ElastiCache tampoco puede usar un grupo de seguridad que se haya modificado.	ELC action: create new security groups

Configuración actual del grupo de seguridad de ElastiCache	Configuración del grupo de seguridad de EC2 actual	Acción de ElastiCache
o cambiar el puerto de una regla existente.		
Hay uno o más grupos de seguridad asociados a la caché de ElastiCache con un nombre que coincide con el patrón `elasticache-ec2-${cacheId}:${ec2InstanceId}` . No se ha modificado ningún grupo de seguridad que coincida con el patrón. Este grupo de seguridad tiene solo una regla de entrada con el grupo de seguridad de VPC de la instancia de EC2 como origen.	Hay uno o más grupos de seguridad asociados a la instancia de EC2 con un nombre que coincide con el patrón `elasticache-ec2-${cacheId}:${ec2InstanceId}` . Sin embargo, ElastiCache no puede usar ninguno de estos grupos de seguridad para la conexión con la caché de ElastiCache. ElastiCache no puede usar un grupo de seguridad que no tenga una regla de entrada con el grupo de seguridad de la VPC de la caché de ElastiCache como origen. ElastiCache tampoco puede usar un grupo de seguridad que se haya modificado.	ELC action: create new security groups

Configuración actual del grupo de seguridad de ElastiCache	Configuración del grupo de seguridad de EC2 actual	Acción de ElastiCache
Hay uno o más grupos de seguridad asociados a la caché de ElastiCache con un nombre que coincide con el patrón `elasticache-ec2-${cacheId}:${ec2InstanceId}` . No se ha modificado ningún grupo de seguridad que coincida con el patrón. Este grupo de seguridad tiene solo una regla de entrada con el grupo de seguridad de VPC de la instancia de EC2 como origen.	Existe un grupo de seguridad de EC2 válido para la conexión, pero no está asociado a la instancia de EC2. Este grupo de seguridad tiene un nombre que coincide con el patrón `ec2-elasticache-${ec2InstanceId}:${cacheId}` . No se ha modificado. Solo tiene una regla de salida con el grupo de seguridad de la VPC de la caché de ElastiCache como origen.	ELC action: associate EC2 security group

Configuración actual del grupo de seguridad de ElastiCache	Configuración del grupo de seguridad de EC2 actual	Acción de ElastiCache
Se aplica alguna de las siguientes condiciones: • No hay ningún grupo de seguridad asociado a la caché de ElastiCache con un nombre que coincida con el patrón `elasticache-ec2-${cacheId}:${ec2InstanceId}` . • Hay uno o más grupos de seguridad asociados a la caché de ElastiCache con un nombre que coincide con el patrón `elasticache-ec2-${cacheId}:${ec2InstanceId}` . Sin embargo, ElastiCache no puede usar ninguno de estos grupos de seguridad para la conexión con la instancia de EC2. ElastiCache no puede usar un grupo de seguridad que no tenga una regla de entrada con el grupo de seguridad de la VPC de la instancia de EC2 como origen. ElastiCache tampoco puede usar un grupo de seguridad modificado.	Hay uno o más grupos de seguridad asociados a la instancia de EC2 con un nombre que coincide con el patrón `ec2-elasticache-${ec2InstanceId}:${cacheId}` . No se ha modificado ningún grupo de seguridad que coincida con el patrón. Este grupo de seguridad solo tiene una regla de salida con el grupo de seguridad de la VPC de la caché de ElastiCache como origen.	ELC action: create new security groups

Acción de ElastiCache: crear nuevos grupos de seguridad

ElastiCache realiza las siguientes acciones:

- Crea un nuevo grupo de seguridad que coincide con el patrón `elasticache-ec2-${cacheId}:${ec2InstanceId}`. Este grupo de seguridad tiene una regla de entrada con el grupo de seguridad de VPC de la instancia de EC2 como origen. Este grupo de seguridad está asociado a la caché de ElastiCache y permite que la instancia de EC2 acceda a ella.
- Crea un nuevo grupo de seguridad que coincide con el patrón `elasticache-ec2-${cacheId}:${ec2InstanceId}`. Este grupo de seguridad tiene una regla de salida con el grupo de seguridad de la VPC de la caché de ElastiCache como origen. Este grupo de seguridad está asociado a la instancia de EC2 y permite que la instancia de EC2 envíe tráfico a la caché de ElastiCache.

Acción de ElastiCache: asociar un grupo de seguridad de EC2

ElastiCache asocia el grupo de seguridad de EC2 válido y existente con la instancia de EC2. Este grupo de seguridad permite que la instancia de EC2 envíe tráfico a la caché de ElastiCache.

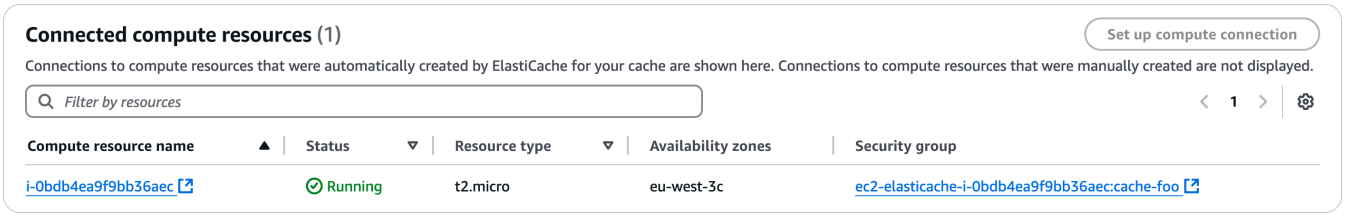
Visualización de los recursos de computación conectados

Puede utilizar la Consola de administración de AWS para ver los recursos de computación que están conectados a una caché de ElastiCache. Los recursos que se muestran incluyen conexiones de recursos informáticos que se configuraron automáticamente. Por ejemplo, puede permitir que un recurso de computación acceda a una caché manualmente añadiendo una regla al grupo de seguridad de la VPC asociado a la caché. Estos recursos no aparecerán en la lista de recursos de computación conectados.

Para que un recurso de computación coincida, se deben cumplir las mismas condiciones que cuando se conectan una instancia de EC2 y una caché de ElastiCache de forma automática.

Cómo visualizar los recursos de computación que están conectados a una caché de ElastiCache

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache.
2. En el panel de navegación, elija Cachés y, a continuación, seleccione una caché de Valkey o Redis OSS.
3. En la pestaña Conectividad y seguridad, consulte los recursos de computación en Recursos de computación conectados.



Escalado de ElastiCache

Puede escalar la caché de ElastiCache para adaptarla a sus necesidades. Las cachés sin servidor y los clústeres basados en nodos ofrecen varias opciones de escalado.

Escalado de ElastiCache Serverless

ElastiCache Serverless acomoda automáticamente el tráfico de carga de trabajo a medida que aumenta o disminuye. En todas las cachés de ElastiCache Serverless, ElastiCache hace un seguimiento continuo del uso de recursos como la CPU, la memoria y la red. Cuando alguno de estos recursos está limitado, ElastiCache Serverless se escala horizontalmente añadiendo una nueva partición y redistribuyendo los datos en la nueva partición, sin que la aplicación sufra periodos de inactividad. Puede controlar los recursos que consume la memoria caché en CloudWatch supervisando la métrica `BytesUsedForCache` del almacenamiento de datos en la caché y la métrica `ElastiCacheProcessingUnits` (ECPU) para el uso computacional.

Establecimiento de límites de escalado para administrar los costes

A fin de controlar el coste de la caché, puede configurar un uso máximo del almacenamiento de datos en caché y de la métrica ECPU/segundo para su caché. Si lo hace, se asegurará de que el uso de la memoria caché nunca supere el máximo configurado.

Si establece un escalado máximo, es posible que su aplicación disminuya el rendimiento de la caché cuando esta alcance el máximo. Cuando establezca un máximo de almacenamiento de datos en caché y alcance dicho máximo, ElastiCache empezará a expulsar los datos de la caché que tengan un tiempo de vida (TTL) establecido, empleando para ello la lógica LRU. Si no hay datos que se puedan expulsar, las solicitudes para escribir datos adicionales recibirán un mensaje de error por memoria insuficiente (OOM). Cuando establezca un máximo de ECPU/segundo y el uso computacional de la carga de trabajo supere ese valor, ElastiCache empezará a limitar las solicitudes.

Si configura un límite máximo en `BytesUsedForCache` o `ElastiCacheProcessingUnits`, es muy recomendable que configure una alarma de CloudWatch con un valor inferior al límite máximo, para que se le notifique cuando la memoria caché esté funcionando cerca de esos límites. Le recomendamos configurar una alarma al 75 % del límite máximo que haya establecido. Consulte la documentación sobre cómo configurar alarmas de CloudWatch.

Escalado previo con ElastiCache sin servidor

Escalado previo de ElastiCache sin servidor

Con el escalado previo, también denominado precalentamiento, puede establecer los límites mínimos admitidos para la caché de ElastiCache. Puede establecer estos mínimos para las unidades de procesamiento de ElastiCache (ECPU) por segundo o para el almacenamiento de datos. Esto puede resultar útil para prepararse para los eventos de escalado previstos. Por ejemplo, si una empresa de videojuegos prevé multiplicar por cinco el número de inicios de sesión en el primer minuto de lanzamiento de su nuevo juego, puede preparar su caché para este importante aumento de uso.

Puede realizar el escalado previo mediante la consola, la CLI o la API de ElastiCache. ElastiCache sin servidor actualiza las ECPU/segundo disponibles en la caché en 60 minutos y envía una notificación de evento cuando se completa la actualización del límite mínimo.

Funcionamiento del escalado previo

Cuando el límite mínimo de ECPU/segundo o almacenamiento de datos se actualiza mediante la consola, la CLI o la API, ese nuevo límite está disponible en 1 hora. ElastiCache sin servidor admite 30 000 ECPU/segundo en una caché vacía y hasta 90 000 ECPU/segundo cuando se utiliza la característica Lectura desde réplica. ElastiCache sin servidor para Valkey 8.0 puede duplicar las solicitudes por segundo (RPS) admitidas cada 2 o 3 minutos, alcanzando 5 millones de RPS por caché desde cero en menos de 13 minutos, con una latencia de lectura constante de p50 de menos de un milisegundo. Si prevé que un próximo evento de escalado podría superar esta velocidad, le recomendamos que ajuste el número mínimo de ECPU/segundo al máximo de ECPU/segundo que tiene previsto al menos 60 minutos antes del pico. De lo contrario, es posible que la aplicación experimente una latencia elevada y se limiten las solicitudes.

Cuando se complete la actualización del límite mínimo, ElastiCache sin servidor empezará a medir las nuevas ECPU mínimas por segundo o el nuevo almacenamiento mínimo. Esto se realiza incluso si la aplicación no ejecuta las solicitudes en la caché o si el uso del almacenamiento de datos es inferior al mínimo. Cuando se reduce el límite mínimo con respecto a su configuración actual, la

actualización es inmediata, por lo que ElastiCache sin servidor empezará a medir el nuevo límite mínimo inmediatamente.

> **Note**
>
> - Al establecer un límite de uso mínimo, se le cobrará por ese límite incluso si su uso real es inferior. El uso del almacenamiento de datos o ECPU que supere el límite de uso mínimo se cobrará a la tarifa normal. Por ejemplo, si establece un límite de uso mínimo de 100 000 ECPU/segundo, se le cobrará al menos 1,224 USD por hora (utilizando los precios de la ECPU en us-east-1), incluso si su uso es inferior al mínimo establecido.
> - ElastiCache sin servidor admite el escalado mínimo solicitado de forma agregada en la caché. ElastiCache sin servidor también admite un máximo de 30 000 ECPU/segundo por ranura (90 000 ECPU/segundo cuando se utiliza Lectura desde réplica mediante conexiones READONLY). Como práctica recomendada, la aplicación debe garantizar que la distribución de claves entre las ranuras de Valkey o Redis OSS y el tráfico entre las claves sea lo más uniforme posible.

Establecimiento de límites de escalado mediante la consola y la AWS CLI

Establecimiento de límites de escalado mediante la consola de AWS

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija el motor que se ejecuta en la caché que desea modificar.
3. Aparecerá una lista de las cachés que ejecutan el motor elegido.
4. Elija la caché que desee modificar seleccionando el botón de opción (a la izquierda del nombre de la caché).
5. Elija Actions (Acciones) y después Modify (Modificar).
6. En Límites de uso, establezca los límites de Memoria o Computación adecuados.
7. Haga clic en Vista previa de los cambios y seleccione Guardar los cambios.

Establecimiento de límites de escalado mediante la AWS CLI

Para cambiar los límites de escalado mediante la CLI, utilice la API modify-serverless-cache.

Linux:

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> \
--cache-usage-limits 'DataStorage={Minimum=10,Maximum=100,Unit=GB},
 ECPUPerSecond={Minimum=1000,Maximum=100000}'
```

Windows:

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> ^
--cache-usage-limits 'DataStorage={Minimum=10,Maximum=100,Unit=GB},
 ECPUPerSecond={Minimum=1000,Maximum=100000}'
```

Eliminación de los límites de escalado mediante la CLI

Para eliminar los límites de escalado mediante la CLI, establezca los parámetros Límite máximo y Límite mínimo en 0.

Linux:

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> \
--cache-usage-limits 'DataStorage={Minimum=0,Maximum=0,Unit=GB},
 ECPUPerSecond={Minimum=0,Maximum=0}'
```

Windows:

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> ^
--cache-usage-limits 'DataStorage={Minimum=0,Maximum=0,Unit=GB},
 ECPUPerSecond={Minimum=0,Maximum=0}'
```

Escalado de clústeres basados en nodos

La cantidad de datos que necesita su aplicación para procesar casi nunca es fija. Aumenta y disminuye a medida que su negocio crece o experimenta las fluctuaciones normales de la demanda. Si administra por sí mismo su caché, necesita aprovisionar suficiente hardware para los picos de demanda, lo cual puede resultar caro. Al utilizar Amazon ElastiCache, puede escalar para satisfacer la demanda actual, al solo pagar por lo que utilice. ElastiCache permite escalar la memoria caché para satisfacer la demanda.

> **Note**
>
> Si un clúster de Valkey o Redis OSS se replica en una o más regiones, esas regiones se escalan en orden. Al escalarse verticalmente, las regiones secundarias se escalan primero y, después, la región principal. Al reducirse verticalmente, la región principal es la primera y luego van las regiones secundarias.
> Al actualizar la versión del motor, primero va la región secundaria y, a continuación, la región principal.

Temas

- Escalado de clústeres para Memcached bajo demanda
- Escalado manual para clústeres de Memcached
- Escalado de clústeres para Valkey o Redis OSS (modo de clúster deshabilitado)
- Escalado de nodos de réplica para Valkey o Redis OSS (modo de clúster deshabilitado)
- Escalado de clústeres de Valkey o Redis OSS (modo de clúster habilitado)

Escalado de clústeres para Memcached bajo demanda

ElastiCache para Memcached ofrece un servicio de almacenamiento en memoria caché totalmente administrado que implementa, opera y escala verticalmente Memcached en la nube de AWS.

Escalado vertical bajo demanda

El escalado vertical de ElastiCache para Memcached es un sistema de almacenamiento en caché de memoria distribuido de alto rendimiento que se utiliza ampliamente para acelerar las aplicaciones dinámicas pues alivian la carga de la base de datos. Los datos y objetos se almacenan en la RAM, lo que reduce la necesidad de leer orígenes de datos externos.

Puede aplicar el escalado vertical tanto a los clústeres basados en nodos existentes como a los nuevos. Esto puede proporcionar flexibilidad a la hora de asignar recursos, lo que permite a los usuarios adaptarse de manera eficiente a las cargas de trabajo cambiantes sin tener que alterar la arquitectura del clúster. Esta capacidad de escalado mejora el rendimiento al aumentar la capacidad de caché durante los períodos de alta demanda y reducirla verticalmente para optimizar los costos durante los períodos de poca demanda. Además, simplifica las operaciones, elimina la necesidad de crear nuevos clústeres para satisfacer los cambios en los requisitos de los recursos y permite responder rápidamente a las fluctuaciones del tráfico. En general, el escalado vertical

de los clústeres basados en nodos para Memcached puede ayudar a aumentar la rentabilidad, mejorar la utilización de los recursos e incluso permitir a los usuarios cambiar el tipo de instancia de Memcached. Todo ello facilita a los usuarios alinear su infraestructura de almacenamiento en caché con las necesidades reales de las aplicaciones.

> **Note**
>
> - Las modificaciones del tipo de nodo solo están disponibles para los clústeres de Memcached basados en nodos con la versión 1.5 o posterior del motor.
> - La detección automática debe estar habilitada para poder utilizar el escalado vertical.

Configuración del escalado vertical bajo demanda para clústeres de Memcached basados en nodos

Puede configurar el escalado vertical bajo demanda para Memcached con `scale-config`, que contiene dos parámetros:

1. ScaleIntervalMinutes: tiempo (en minutos) entre el escalado de los lotes durante el proceso de actualización de Memcached
2. ScalePercentage: porcentaje de nodos que se escalará de forma simultánea durante el proceso de actualización de Memcached

Conversión de un tipo de nodo de Memcached existente en una caché que pueda escalarse verticalmente desde la CLI

Para convertir un clúster basado en nodos de Memcached existente en una caché que pueda escalarse verticalmente, puede usar `elasticache modify-cache-cluster` en la CLI.

```
aws elasticache modify-cache-cluster \
    --cache-cluster-id <your-cluster-id> \
    --cache-node-type <new-node-type> \
    --scale-config <scale-config> \
    --apply-immediately
```

Configuración del escalado vertical con la CLI

Para configurar el escalado vertical de un clúster de Memcached basado en nodos mediante la CLI, utilice `elasticache modify-cache-cluster` con `scale-config` y sus parámetros `ScalePercentage` y `ScaleIntervalMinutes`.

- scale-interval-minutes: define el tiempo (en minutos) entre lotes de escalado. Este valor puede definirse entre 2 y 30 minutos. Si no se especifica ningún valor, se utiliza el valor predeterminado de 5 minutos.
- scale-percentage: especifica el porcentaje de nodos que se escalará simultáneamente en cada lote. Este ajuste puede definirse entre 10 y 100 minutos. El valor se redondea al alza al dividir, por lo que si, por ejemplo, el resultado es 49,5, se aplica una configuración de 50. Si no se especifica ningún valor, se utiliza el valor predeterminado de 20.

Estas opciones de configuración le permitirán refinar el proceso de escalado de acuerdo con sus necesidades específicas, estableciendo un equilibro entre minimizar las interrupciones del clúster y optimizar la velocidad de escalado. El parámetro scale-config solo se aplicará a los tipos de motor de Memcached y se omitirá en el caso de otros motores de caché, lo que garantizará la compatibilidad con el uso de las API existente en otros clústeres.

Llamada a la API

```
aws elasticache modify-cache-cluster \
    --cache-cluster-id <your-cluster-id> \
    --cache-node-type <new-node-type> \
    --scale-config '{
            "ScalePercentage": 30,
            "ScaleIntervalMinutes": 2
          }'
    --apply-immediately
```

Resultado:

Devuelve el ID del clúster y el cambio pendiente.

```
{
    "CacheCluster": {
        "CacheNodeType": "old_insance_type",
         ...
         ...
         "PendingModifiedValues": {
            "CacheNodeType": "new_instance_type"
         },
    }
}
```

Enumeración de la configuración de escalado vertical de su caché de Memcached

Puede recuperar las opciones de escalado de sus cachés de Memcached y ver cuáles son sus opciones actuales de escalado vertical.

Llamada a la API

```
aws elasticache list-allowed-node-type-modifications --cache-cluster-id <your-cluster-id>
```

Resultado:

```
{
  "ScaleUpModifications": [
      "cache.x.xxxx",
      "cache.x.xxxx"
      ],
   "ScaleDownModifications": [
      "cache.x.xxxx",
      "cache.x.xxxx",
      "cache.x.xxxx"
      ]
}
```

Escalado vertical para Memcached con la Consola de administración de AWS

Siga estos pasos para utilizar la Consola de administración de AWS para convertir un clúster de Memcached basado en nodos en un clúster escalable verticalmente.

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. Seleccione el clúster de Memcached que desee convertir.
3. Seleccione la pestaña Modificar.
4. Vaya a la sección Configuración de la caché y seleccione el tipo de nodo deseado.
5. Seleccione Vista previa de los cambios y revise los cambios.
6. Seleccione Modificar.

Escalado horizontal automatizado para Memcached

ElastiCache ahora está integrado con el servicio Escalado automático de aplicaciones de AWS (AAS) para incluir el escalado horizontal automatizado para los clústeres de Memcached. Puede definir políticas de escalado a través del servicio Escalado automático de aplicaciones de AWS y ajustar automáticamente la cantidad de nodos en los clústeres de Memcached según sea necesario, en función de las métricas o programaciones predefinidas.

> **Note**
>
> El escalado horizontal automatizado no está disponible actualmente en las regiones de Pekín y Ningxia.

Estos son los métodos disponibles para escalar automáticamente de forma horizontal los clústeres basados en nodos.

- Escalado programado: el escalado basado en una programación le permite definir su propia programación de escalado para los cambios en las cargas predecibles. Por ejemplo, cada semana el tráfico de su aplicación web empieza a aumentar los miércoles, sigue siendo alto el jueves y comienza a disminuir el viernes. Puede configurar el escalado automático para aumentar la capacidad los miércoles y reducirla los viernes.
- Seguimiento de destinos: las políticas de escalado de seguimiento de destinos le permiten elegir una métrica de escalado y establecer un valor de destino. Auto Scaling de aplicaciones crea y administra las alarmas de CloudWatch que desencadenan la política de escalado y calcula el ajuste de escalado según la métrica y el valor de destino. La política de escalado amplía o reduce la capacidad en función de las necesidades para mantener la métrica en el valor objetivo especificado o en un valor próximo.

Cómo configurar el escalado horizontal para un clúster de Memcached basado en nodos mediante la CLI

Al escalar horizontalmente un clúster de Memcached basado en nodos, puede utilizar una política de seguimiento de destinos, una política programada o ambas.

1. Registro de un recurso como destino escalable

Llame a la API `RegisterScalableTarget` en Escalado automático de aplicaciones de AWS para registrar el destino de la dimensión escalable `elasticache:cache-cluster:Nodes`.

API: ApplicationAutoScaling.RegisterScalableTarget

Input:

```
{
 "ScalableDimension": "elasticache:cache-cluster:Nodes",
 "ResourceId": "cache-cluster/test-cluster-1",
 "ServiceNamespace": "elasticache",
 "MinCapacity": 20,
 "MaxCapacity": 50
}
```

2. Creación de una política de escalado de seguimiento de destino

A continuación, puede crear una política de escalado de seguimiento de destino para el recurso llamando a la API de la política de escalado.

3. Métrica predefinida

A continuación se muestra una política que aplica el escalado a la dimensión del nodo de caché con la métrica predefinida `ElastiCacheCPUUtilization`, establecida en 50 para el clúster test-cluster-1. Al eliminar nodos para reducirlos horizontalmente, se borran los últimos n nodos.

API: ApplicationAutoScaling.PutScalingPolicy

Input:

```
{
 "PolicyName": "cpu50-target-tracking-scaling-policy",
 "PolicyType": "TargetTrackingScaling",
 "TargetTrackingScalingPolicyConfiguration": {
  "TargetValue": 50,
  "PredefinedMetricSpecification": {
   "PredefinedMetricType": "ElastiCacheCPUUtilization"
   },
  "ScaleOutCooldown": 600,
  "ScaleInCooldown": 600
   },
 "ServiceNamespace": "elasticache",
```

```
 "ScalableDimension": "elasticache:cache-cluster:Nodes",
 "ResourceId": "cache-cluster/test-cluster-1"
}
```

Salida:

```
{
 "PolicyARN": "arn:aws:autoscaling:us-west-2:012345678910:scalingPolicy:6d8972f3-
efc8-437c-92d1-6270f29a66e7:resource/elasticache/cache-cluster/test-
cluster-1:policyName/cpu50-target-tracking-scaling-policy",
 "Alarms": [
  {
  "AlarmARN": "arn:aws:cloudwatch:us-west-2:012345678910:alarm:TargetTracking-
elasticache/cache-cluster/test-cluster-1-AlarmHigh-d4f0770c-b46e-434a-
a60f-3b36d653feca",
  "AlarmName": "TargetTracking-elasticache/cache-cluster/test-cluster-1-AlarmHigh-
d4f0770c-b46e-434a-a60f-3b36d653feca"
  },
  {
  "AlarmARN": "arn:aws:cloudwatch:us-west-2:012345678910:alarm:TargetTracking-
elasticache/cache-cluster/test-cluster-1-AlarmLow-1b437334-d19b-4a63-
a812-6c67aaf2910d",
  "AlarmName": "TargetTracking-elasticache/cache-cluster/test-cluster-1-
AlarmLow-1b437334-d19b-4a63-a812-6c67aaf2910d"
  }
 ]
}
```

4. Métricas personalizadas

   También puede establecer una política de escalado en función de la dimensión usando un porcentaje personalizado que se base en la métrica de Cloudwatch.

   Input:

```
{
 "PolicyName": "cpu50-target-tracking-scaling-policy",
 "PolicyType": "TargetTrackingScaling",
 "TargetTrackingScalingPolicyConfiguration": {
  "CustomizedMetricSpecification": {
   "Dimensions": [
    {
    "Name": "MyMetricDimension",
```

```
    "Value": "DimensionValue"
    }
    ],
   "MetricName": "MyCustomMetric",
   "Namespace": "MyNamespace",
   "Statistic": "Average",
   "Unit": "Percent"
   },
  "TargetValue": 40,
  "ScaleOutCooldown": 600,
  "ScaleInCooldown": 600
  },
 "ServiceNamespace": "elasticache",
 "ScalableDimension": "elasticache:cache-cluster:Nodes",
 "ResourceId": "cache-cluster/test-cluster-1"
}
```

5. Acciones programadas

   Cuando tenga que escalar horizontalmente para un evento determinado y, después, reducir horizontalmente, puede crear dos acciones programadas llamando a la API `PutScheduledAction`.

   Política 1: escalado horizontal

   El comando `at` en `--schedule` programa la acción que se va a ejecutar una sola vez en una fecha y hora específicas en el futuro. El campo de programación también admite frecuencias (minuto, hora, día, etc.) y cron (para la expresión cron).

   En la fecha y la hora especificadas, Escalado automático de aplicaciones actualiza los valores `MinCapacity` y `MaxCapacity`. El escalado automático de aplicaciones se escala horizontalmente a MinCapacity para poner los nodos de caché en 70.

   API: ApplicationAutoScaling.PutScheduledAction

   Input:

```
{
 "ResourceId": "elasticache:ache-cluster:test-cluster-1",
 "ScalableDimension": "elasticache:cache-cluster:Nodes",
  "ScalableTargetAction": {
   "MaxCapacity": 100,
```

```
   "MinCapacity": 70
   },
 "Schedule": "at(2020-05-20T17:05:00)",
 "ScheduledActionName": "ScalingOutScheduledAction",
 "ServiceNamespace": "elasticache",
}
```

Política 2: reducción horizontal

En la fecha y la hora especificadas, Escalado automático de aplicaciones actualiza los valores `MinCapacity` y `MaxCapacity` de la tabla y reduce horizontalmente a `MaxCapacity` para devolver los nodos de caché a 60.

API: ApplicationAutoScaling.PutScheduledAction

Input:

```
{
 "ResourceId": "elasticache:cache-cluster:test-cluster-1",
 "ScalableDimension": "elasticache:cache-cluster:Nodes",
 "ScalableTargetAction": {
  "MaxCapacity": 60,
  "MinCapacity": 40
  },
 "Schedule": "at(2020-05-21T17:05:00)",
 "ScheduledActionName": "ScalingInScheduledAction",
 "ServiceNamespace": "elasticache",
}
```

6. Visualización de las actividades de escalado

Puede ver las actividades de escalado mediante la API `DescribeScalingActivities`.

API: ApplicationAutoScaling.DescribeScalingActivities

Salida:

```
{
 "ScalingActivities": [
  {
  "ScalableDimension": "elasticache:elasticache:DesiredCount",
  "Description": "Setting desired count to 30.",
```

```
  "ResourceId": "elasticache/cache-cluster/test-cluster-1",
  "ActivityId": "4d759079-a31f-4d0c-8468-504c56e2eecf",
  "StartTime": 1462574194.658,
  "elasticacheNamespace": "elasticache",
  "EndTime": 1462574276.686,
  "Cause": "monitor alarm TargetTracking-elasticache/cache-cluster/test-cluster-1-
AlarmHigh-d4f0770c-b46e-434a-a60f-3b36d653feca in state ALARM triggered policy
 cpu50-target-tracking-scaling-policy",
  "StatusMessage": "Failed to set desired count to 30",
  "StatusCode": "Failed"
  },
  {
  "ScalableDimension": "elasticache:elasticache:DesiredCount",
  "Description": "Setting desired count to 25.",
  "ResourceId": "elasticache/cache-cluster/test-cluster-1",
  "ActivityId": "90aff0eb-dd6a-443c-889b-b809e78061c1",
  "StartTime": 1462574254.223,
  "elasticacheNamespace": "elasticache",
  "EndTime": 1462574333.492,
  "Cause": "monitor alarm TargetTracking-elasticache/cache-cluster/test-cluster-1-
AlarmHigh-d4f0770c-b46e-434a-a60f-3b36d653feca in state ALARM triggered policy
 cpu50-target-tracking-scaling-policy",
  "StatusMessage": "Successfully set desired count to 25. Change successfully
 fulfilled by elasticache.",
  "StatusCode": "Successful"
  }
 ]
}
```

7. Edición o eliminación de la política de escalado

   Para editar o eliminar políticas, vuelva a llamar a la API `PutScalingPolicy` o llame a la acción `DeleteScalingPolicy` o `DeleteScheduled`.

8. Anulación del registro de los destinos escalables

   Puede anular el registro del destino escalable a través de la API `DeregisterScalableTarget`. Al anular el registro de un destino escalable, se eliminan las políticas de escalado y las acciones programadas asociadas a él.

   API: ApplicationAutoScaling.DeregisterScalableTarget

   Input:

```
{
 "ResourceId": "elasticache/cache-cluster/test-cluster-1",
 "ServiceNamespace": "elasticache",
 "ScalableDimension": "elasticache:cache-cluster:Nodes"
}
```

9. Limpieza de las políticas de escalado
10. Políticas de escalado múltiples

    Puede crear varias políticas de escalado. A continuación, se muestran los principales comentarios sobre el comportamiento del Seguimiento de destinos de escalado automático.

    - Puede tener varias políticas de escalado de seguimiento de destino para un destino escalable, siempre que cada una de ellas utilice una métrica diferente.
    - El objetivo de Auto Scaling de aplicaciones siempre es dar prioridad a la disponibilidad, por lo que su comportamiento varía en rol de si las políticas de seguimiento de destino están listas para el escalado horizontal o reducción horizontal. Realizará un escalado ascendente del destino escalable si cualquiera de las políticas de seguimiento de destino está lista para el escalado ascendente, pero solo realizará el escalado descendente si todas las políticas de seguimiento de destino (que tienen la parte de escalado descendente habilitada) están listas para el escalado descendente.
    - Si varias políticas indican al destino escalable que escale horizontalmente o reduzca horizontalmente al mismo tiempo, Auto Scaling de aplicaciones se escala en función de la política que proporciona la mayor capacidad para escalar horizontalmente y reducir horizontalmente Esto brinda una mayor flexibilidad para abordar diferentes situaciones y garantizar que siempre haya capacidad suficiente para procesar las cargas de trabajo de la aplicación.

    > **Note**
    >
    > El Escalado automático de aplicaciones de AWS no pone las políticas de escalado en cola. El escalado automático de aplicaciones esperará a que se complete el primer escalado, luego descansará y, a continuación, repetirá el algoritmo anterior.

Escalado horizontal automático de un clúster de Memcached basado en nodos desde la Consola de administración de AWS

Siga estos pasos para utilizar la Consola de administración de AWS para convertir un clúster de Memcached basado en nodos en un clúster escalable verticalmente.

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. Seleccione la caché de Memcached que desee convertir.
3. Vaya a la pestaña Escalado automático.
4. Para elegir la política de escalado que desee aplicar, seleccione Agregar escalado dinámico o Agregar escalado programado.
5. Rellene los detalles de la política seleccionada según sea necesario.
6. Haga clic en Crear.

Escalado manual para clústeres de Memcached

Escalar un clúster de Memcached horizontal y verticalmente es tan sencillo como añadir o eliminar nodos del clúster. Los clústeres de Memcached se componen de 1 a 60 nodos.

Dado que puede fragmentar los datos entre todos los nodos de un clúster de Memcached, el escalado con ampliación a un tipo de nodo con más memoria no suele ser necesaria. Sin embargo, debido a que el motor de Memcached no conserva los datos, si escala a un tipo de nodo distinto, el clúster nuevo estará vacío a menos que la aplicación lo rellene.

Para escalar verticalmente su clúster de Memcached de forma manual, debe crear un clúster nuevo. Los clústeres de Memcached siempre comienzan vacíos a menos que su aplicación los rellene.

Escalado manual de clústeres de Memcached

Acción	Tema
Escalado ascendente	Adición de nodos a un clúster
Escalado descendente	Eliminación de nodos de un clúster
Cambios de tipos de nodos	

Acción	Tema
	Escalado manual de clústeres de Memcached basados en nodos verticalmente

Temas

- Escalado manual de un clúster de Memcached basado en nodos horizontalmente
- Escalado manual de clústeres de Memcached basados en nodos verticalmente

Escalado manual de un clúster de Memcached basado en nodos horizontalmente

El motor de Memcached permite realizar una partición de sus datos entre varios nodos. Por ello, los clústeres de Memcached se pueden escalar de forma horizontal fácilmente. Para escalar horizontalmente su clúster de Memcached, simplemente agregue o elimine nodos.

Los siguientes temas detallan cómo escalar su clúster de Memcached de forma descendente o ascendente al agregar o eliminar nodos.

- Adición de nodos a un clúster
- Eliminación de nodos del clúster

Cada vez que cambie el número de nodos del clúster de Memcached, debe volver a asignar al menos algunos de su espacio de claves para que se asocien al nodo correcto. Para obtener información más detallada sobre cómo equilibrar la carga de su clúster de Memcached, consulte Configuración de su cliente de ElastiCache para un equilibrio de carga eficaz (Memcached).

Si utiliza la detección automática en su clúster de Memcached, no es necesario cambiar los puntos de enlace de su aplicación al agregar o eliminar nodos. Para obtener más información sobre la detección automática, consulte Identificación automática de los nodos en el clúster (Memcached). Si no utiliza la detección automática, cada vez que cambie el número de nodos de su clúster de Memcached, deberá actualizar los puntos de conexión de su aplicación.

Escalado manual de clústeres de Memcached basados en nodos verticalmente

Para escalar o reducir su clúster de Memcached manualmente, debe crear un nuevo clúster. Los clústeres de Memcached siempre comienzan vacíos a menos que su aplicación los rellene.

> **Important**
>
> Si escala para reducir a un tipo de nodo más pequeño, asegúrese de que el tipo de nodo más pequeño sea adecuado para sus datos, con una capacidad adicional. Para obtener más información, consulte Elección del tamaño del nodo.

Temas

- Escalado vertical de clústeres de Memcached basados en nodos (consola)
- Escalado vertical de un clúster de Memcached basado en nodos (AWS CLI)
- Escalado vertical de un clúster de Memcached basado en nodos (API de ElastiCache)

Escalado vertical de clústeres de Memcached basados en nodos (consola)

El siguiente procedimiento le muestra cómo escalar verticalmente su clúster de Memcached basado en nodos utilizando la Consola de administración de AWS.

1. Cree un nuevo clúster con el nuevo tipo de nodo. Para obtener más información, consulte Creación de un clúster de Memcached (consola).
2. En la aplicación, actualice los puntos de enlace a los puntos de enlace del nuevo clúster. Para obtener más información, consulte Búsqueda de puntos de conexión de un clúster (consola) (Memcached).
3. Elimine el clúster anterior. Para obtener más información, consulte Deleting a new node in Memcached.

Escalado vertical de un clúster de Memcached basado en nodos (AWS CLI)

El siguiente procedimiento le muestra cómo escalar verticalmente su clúster de Memcached basado en nodos utilizando la AWS CLI.

1. Cree un nuevo clúster con el nuevo tipo de nodo. Para obtener más información, consulte Creación de un clúster (AWS CLI).
2. En la aplicación, actualice los puntos de enlace a los puntos de enlace del nuevo clúster. Para obtener más información, consulte Búsqueda de puntos de conexión (AWS CLI).
3. Elimine el clúster anterior. Para obtener más información, consulte Uso de AWS CLI para eliminar un clúster de ElastiCache.

Escalado vertical de un clúster de Memcached basado en nodos (API de ElastiCache)

El siguiente procedimiento le muestra cómo escalar verticalmente su clúster de Memcached basado en nodos utilizando la API de ElastiCache.

1. Cree un nuevo clúster con el nuevo tipo de nodo. Para obtener más información, consulte Creación de un clúster para Memcached (API de ElastiCache)
2. En la aplicación, actualice los puntos de enlace a los puntos de enlace del nuevo clúster. Para obtener más información, consulte Búsqueda de puntos de enlace (API de ElastiCache).
3. Elimine el clúster anterior. Para obtener más información, consulte Uso de la API de ElastiCache.

Escalado de clústeres para Valkey o Redis OSS (modo de clúster deshabilitado)

Los clústeres de Valkey o Redis OSS (modo de clúster deshabilitado) pueden tener un solo nodo con 0 particiones o varios nodos con 1 partición. Los clústeres de un solo nodo utilizan el nodo para las lecturas y las escrituras. Los clústeres de varios nodos siempre tienen 1 nodo como nodo primario de lectura/escritura con entre 0 y 5 nodos de réplica de solo lectura.

Temas

- Escalado de clústeres para Valkey o Redis OSS (modo de clúster deshabilitado)

Escalado de clústeres de Valkey o Redis OSS

Acción	Valkey o Redis OSS (modo de clúster deshabilitado)	Valkey o Redis OSS (modo de clúster habilitado)
Escalado descendente	Eliminación de nodos de un clúster de ElastiCache	Escalado de clústeres de Valkey o Redis OSS (modo de clúster habilitado)
Escalado ascendente	Adición de nodos a un clúster	Cambio de particiones en línea para Valkey o Redis OSS (modo de clúster habilitado)
Cambios de tipos de nodos	A un tipo de nodo más grande: • Escalado vertical de clústeres de Valkey o Redis OSS de un solo nodo • Escalado vertical de clústeres de Valkey o Redis OSS con réplicas A un tipo de nodo más pequeño:	Escalado vertical en línea mediante la modificación del tipo de nodo

Acción	Valkey o Redis OSS (modo de clúster deshabilitado)	Valkey o Redis OSS (modo de clúster habilitado)
	• Reducción vertical de clústeres de Valkey o Redis OSS de un solo nodo • Reducción vertical de clústeres de Valkey o Redis OSS con réplicas	
Cambio del número de grupos de nodos	No es compatible con los clústeres de Valkey o Redis OSS (modo de clúster deshabilitado)	Escalado de clústeres de Valkey o Redis OSS (modo de clúster habilitado)

Contenido

- Escalado de clústeres para Valkey o Redis OSS (modo de clúster deshabilitado)
  - Escalado vertical de clústeres de Valkey o Redis OSS de un solo nodo
    - Escalado vertical de clústeres de Valkey o Redis OSS (modo de clúster deshabilitado) de un solo nodo (consola)
    - Escalado vertical de clústeres de Valkey o Redis OSS de un solo nodo (AWS CLI)
    - Escalado vertical de clústeres de Valkey o Redis OSS de un solo nodo (API de ElastiCache)
  - Reducción vertical de clústeres de Valkey o Redis OSS de un solo nodo
    - Reducción vertical de clústeres de Valkey o Redis OSS de un solo nodo (consola)
    - Reducción vertical de clústeres de Valkey o Redis OSS de un solo nodo (AWS CLI)
    - Reducción vertical de clústeres de Valkey o Redis OSS de un solo nodo (API de ElastiCache)

Escalado de clústeres para Valkey o Redis OSS (modo de clúster deshabilitado)

Los nodos de Valkey o Redis OSS (modo de clúster deshabilitado) deben ser lo suficientemente grandes como para contener todos los datos de la caché, además de asumir la sobrecarga de Valkey o Redis OSS. Para cambiar la capacidad de los datos de su clúster de Valkey o Redis OSS (modo de clúster deshabilitado), debe realizar un escalado vertical; es decir, escalar verticalmente a un tipo de

nodo más grande para aumentar la capacidad de los datos o reducir verticalmente a un tipo de nodo más pequeño para reducir la capacidad de los datos.

El proceso de escalado vertical de ElastiCache se ha diseñado para hacer todo lo posible por retener sus datos y necesita que la replicación de Valkey o Redis OSS se realice correctamente. Para clústeres de Valkey o Redis OSS (modo de clúster deshabilitado), recomendamos disponer de suficiente memoria para Valkey o Redis OSS.

No puede particionar los datos entre varios clústeres de Valkey o Redis OSS (modo de clúster deshabilitado). Sin embargo, si solo necesita aumentar o reducir la capacidad de lectura de su clúster, puede crear un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) con nodos de réplica y agregar o eliminar réplicas de lectura. Para crear un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) con nodos de réplica mediante su clúster de Valkey o Redis OSS de un único nodo como el clúster principal, consulte Creación de un clúster de Valkey (modo de clúster deshabilitado) (consola).

Después de crear el clúster con réplicas, puede aumentar la capacidad de lectura agregando réplicas de lectura. Después, si lo necesita, puede reducir la capacidad de lectura quitando réplicas de lectura. Para obtener más información, consulte Aumento de la capacidad de lectura o Reducción de la capacidad de lectura.

Además de poder escalar la capacidad de lectura, los clústeres de Valkey o Redis OSS (modo de clúster deshabilitado) con réplicas tienen otras ventajas empresariales. Para obtener más información, consulte Alta disponibilidad a través de grupos de reproducción.

> **Important**
>
> Si su grupo de parámetros usa `reserved-memory` para apartar memoria para la capacidad adicional de Valkey o Redis OSS, antes de comenzar el escalado, asegúrese de tener un grupo de parámetros personalizado que reserve la cantidad de memoria adecuada para el nuevo tipo de nodo. De forma alternativa, puede modificar un grupo de parámetros personalizado para que use `reserved-memory-percent` y utilizar ese grupo de parámetros para su nuevo clúster.
> Si va a utilizar `reserved-memory-percent`, esto no es necesario.
> Para obtener más información, consulte Administración de la memoria reservada para Valkey y Redis OSS.

Temas

- Escalado vertical de clústeres de Valkey o Redis OSS de un solo nodo
- Reducción vertical de clústeres de Valkey o Redis OSS de un solo nodo

Escalado vertical de clústeres de Valkey o Redis OSS de un solo nodo

Al escalar verticalmente un clúster de Valkey o Redis OSS de un solo nodo, ElastiCache realiza el siguiente proceso, independientemente de si utiliza la consola de ElastiCache, la AWS CLI o la API de ElastiCache.

1. Un nuevo clúster con el nuevo tipo de nodo se activa en la misma zona de disponibilidad que el clúster existente.
2. Los datos de caché del clúster de caché existente se copian en el nuevo clúster. El tiempo que tarda este proceso depende de su tipo de nodo y de la cantidad de datos que haya en el clúster.
3. Las operaciones de lectura y escritura se sirven ahora utilizando el nuevo clúster. Puesto que los puntos de conexión del nuevo clúster son los mismos que los que había en el clúster anterior, no es necesario actualizar los puntos de conexión de su aplicación. Observará una breve interrupción (algunos segundos) de las operaciones de lectura y escritura del nodo primario mientras se actualiza la entrada de DNS.
4. ElastiCache elimina el clúster anterior. Observará una breve interrupción (unos segundos) de las operaciones de lectura y escritura del nodo anterior porque se desconectarán las conexiones con el nodo anterior.

> **Note**
>
> Para los clústeres que ejecutan el tipo de nodo r6gd, solo puede escalar a tamaños de nodo dentro de la familia de nodos r6gd.

Tal y como se muestra en la siguiente tabla, la operación de escalado vertical de Valkey o Redis OSS se bloqueará si tiene una actualización de motor programada para el próximo periodo de mantenimiento. Para obtener más información acerca de los periodos de mantenimiento, consulte Administración del mantenimiento de clústeres de ElastiCache.

Operaciones bloqueadas de Valkey o Redis OSS

Operaciones pendientes	Operaciones bloqueadas
Escalado ascendente	Actualización del motor inmediata
Actualización del motor	Ampliación inmediata

Operaciones pendientes	Operaciones bloqueadas
Ampliación y actualización del motor	Ampliación inmediata
	Actualización del motor inmediata

Si hay una operación pendiente que esté bloqueando, tiene las siguientes opciones.

- Programe la operación de escalado vertical de Valkey o Redis OSS para el siguiente periodo de mantenimiento desactivando la casilla de verificación Aplicar inmediatamente (en la CLI: use `--no-apply-immediately`, en la API: use `ApplyImmediately=false`).
- Espere al siguiente periodo de mantenimiento (o después) para realizar la operación de escalado vertical de Valkey o Redis OSS.
- Añada la actualización del motor de Valkey o Redis OSS a esta modificación de clúster con la casilla de verificación Aplicar inmediatamente activada (en la CLI, use `--apply-immediately` y en la API, `ApplyImmediately=true`). Esto desbloqueará su operación de ampliación, ya que hará que la actualización del motor se realice inmediatamente.

Puede escalar verticalmente un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) de un solo nodo con la consola de ElastiCache, la AWS CLI o la API de ElastiCache.

> **Important**
>
> Si su grupo de parámetros usa `reserved-memory` para apartar memoria para la capacidad adicional de Valkey o Redis OSS, antes de comenzar el escalado, asegúrese de tener un grupo de parámetros personalizado que reserve la cantidad de memoria adecuada para el nuevo tipo de nodo. De forma alternativa, puede modificar un grupo de parámetros personalizado para que use `reserved-memory-percent` y utilizar ese grupo de parámetros para su nuevo clúster.
> Si va a utilizar `reserved-memory-percent`, esto no es necesario.
> Para obtener más información, consulte Administración de la memoria reservada para Valkey y Redis OSS.

Escalado vertical de clústeres de Valkey o Redis OSS (modo de clúster deshabilitado) de un solo nodo (consola)

El siguiente procedimiento describe cómo escalar verticalmente un clúster de Valkey o Redis OSS de un solo nodo mediante la consola de administración de ElastiCache. Durante este proceso, el clúster de Valkey o Redis OSS seguirá atendiendo solicitudes con un tiempo de inactividad mínimo.

Escalado vertical de un clúster de Valkey o Redis OSS de un solo nodo (consola)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Clústeres de Valkey o Redis OSS.
3. En la lista de clústeres, elija el clúster que desee escalar verticalmente (debe ejecutar el motor de Valkey o Redis OSS, no el motor de Valkey o Redis OSS en clúster).
4. Elija Modificar.
5. En el asistente Modify Cluster:
   a. Elija el tipo de nodo que desee ampliar en la lista Node type.
   b. Si va a utilizar `reserved-memory` para administrar su memoria, en la lista Parameter Group, elija el grupo de parámetros personalizado que reserve la cantidad correcta de memoria para su nuevo tipo de nodo.
6. Si desea realizar el proceso de ampliación de forma inmediata, elija el cuadro Apply immediately. Si no selecciona el cuadro Apply immediately, el proceso de ampliación se realizará durante el siguiente periodo de mantenimiento de este clúster.
7. Elija Modify (Modificar).

   Si eligió Apply immediately (Aplicar inmediatamente) en el paso anterior, el estado del clúster cambiará a modifying (en modificación). Cuando el estado cambie a available (disponible), la modificación se habrá completado y podrá empezar a utilizar el nuevo clúster.

Escalado vertical de clústeres de Valkey o Redis OSS de un solo nodo (AWS CLI)

El siguiente procedimiento describe cómo escalar verticalmente un clúster de Valkey o Redis OSS de un solo nodo mediante la AWS CLI. Durante este proceso, el clúster de Valkey o Redis OSS seguirá atendiendo solicitudes con un tiempo de inactividad mínimo.

Cómo escalar verticalmente un clúster de Valkey o Redis OSS de un solo nodo (AWS CLI)

1. Determine los tipos de nodos a los que puede ampliar ejecutando el comando AWS CLI de la `list-allowed-node-type-modifications` con el siguiente parámetro.

   - `--cache-cluster-id`

   Para Linux, macOS o Unix:

   ```
   aws elasticache list-allowed-node-type-modifications \
        --cache-cluster-id my-cache-cluster-id
   ```

   Para Windows:

   ```
   aws elasticache list-allowed-node-type-modifications ^
        --cache-cluster-id my-cache-cluster-id
   ```

   La salida del comando anterior es similar a la siguiente (formato JSON).

   ```
   {
        "ScaleUpModifications": [
            "cache.m3.2xlarge",
            "cache.m3.large",
            "cache.m3.xlarge",
            "cache.m4.10xlarge",
            "cache.m4.2xlarge",
            "cache.m4.4xlarge",
            "cache.m4.large",
            "cache.m4.xlarge",
            "cache.r3.2xlarge",
            "cache.r3.4xlarge",
            "cache.r3.8xlarge",
            "cache.r3.large",
            "cache.r3.xlarge"
        ]
           "ScaleDownModifications": [
            "cache.t2.micro",
            "cache.t2.small ",
            "cache.t2.medium ",
               "cache.t1.small ",
        ],
   ```

```
}
```

Para obtener más información consulte list-allowed-node-type-modifications en la Referencia de la AWS CLI.

2. Modifique el clúster existente especificando el clúster que se debe escalar verticalmente e indique el nuevo tipo de nodo más grande con el comando `modify-cache-cluster` de la AWS CLI y los siguientes parámetros.

   - `--cache-cluster-id`: nombre del clúster que está escalando verticalmente.
   - `--cache-node-type`: tipo de nodo nuevo al que desea escalar el clúster. Este valor debe ser uno de los tipos de nodos devueltos por el comando `list-allowed-node-type-modifications` en el paso 1.
   - `--cache-parameter-group-name`: [Opcional] Utilice este parámetro si va a utilizar `reserved-memory` para administrar la memoria reservada de su clúster. Especifique un grupo de parámetros de caché personalizado que reserve la cantidad correcta de memoria para el nuevo tipo de nodo. Si va a utilizar `reserved-memory-percent`, puede omitir este parámetro.
   - `--apply-immediately`: hace que el proceso de escalado vertical se aplique de inmediato. Para aplazar el proceso de ampliación al siguiente periodo de mantenimiento del clúster, utilice el parámetro `--no-apply-immediately`.

   Para Linux, macOS o Unix:

   ```
   aws elasticache modify-cache-cluster \
        --cache-cluster-id my-redis-cache-cluster \
        --cache-node-type cache.m3.xlarge \
        --cache-parameter-group-name redis32-m2-xl \
        --apply-immediately
   ```

   Para Windows:

   ```
   aws elasticache modify-cache-cluster ^
        --cache-cluster-id my-redis-cache-cluster ^
        --cache-node-type cache.m3.xlarge ^
        --cache-parameter-group-name redis32-m2-xl ^
        --apply-immediately
   ```

La salida del comando anterior es similar a la siguiente (formato JSON).

```
{
     "CacheCluster": {
         "Engine": "redis",
         "CacheParameterGroup": {
             "CacheNodeIdsToReboot": [],
             "CacheParameterGroupName": "default.redis6.x",
             "ParameterApplyStatus": "in-sync"
         },
         "SnapshotRetentionLimit": 1,
         "CacheClusterId": "my-redis-cache-cluster",
         "CacheSecurityGroups": [],
         "NumCacheNodes": 1,
         "SnapshotWindow": "00:00-01:00",
         "CacheClusterCreateTime": "2017-02-21T22:34:09.645Z",
         "AutoMinorVersionUpgrade": true,
         "CacheClusterStatus": "modifying",
         "PreferredAvailabilityZone": "us-west-2a",
         "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
         "CacheSubnetGroupName": "default",
         "EngineVersion": "6.0",
         "PendingModifiedValues": {
             "CacheNodeType": "cache.m3.2xlarge"
         },
         "PreferredMaintenanceWindow": "tue:11:30-tue:12:30",
         "CacheNodeType": "cache.m3.medium",
          "DataTiering": "disabled"
     }
 }
```

Para obtener más información consulte modify-cache-cluster en la Referencia de la AWS CLI.

3. Si ha utilizado `--apply-immediately`, compruebe el estado del nuevo clúster con el comando `describe-cache-clusters` de la AWS CLI con el siguiente parámetro. Cuando el estado cambie a disponible, podrá comenzar a usar el nuevo clúster de mayor tamaño.

   - `--cache-cluster-id`: nombre del clúster de Valkey o Redis OSS de un solo nodo. Use este parámetro para describir un clúster determinado en lugar de todos los clústeres.

```
aws elasticache describe-cache-clusters --cache-cluster-id my-redis-cache-cluster
```

Para obtener más información consulte describe-cache-clusters en la Referencia de la AWS CLI.

Escalado vertical de clústeres de Valkey o Redis OSS de un solo nodo (API de ElastiCache)

El siguiente procedimiento describe cómo escalar verticalmente un clúster de Valkey o Redis OSS de un solo nodo mediante la API de ElastiCache. Durante este proceso, el clúster de Valkey o Redis OSS seguirá atendiendo solicitudes con un tiempo de inactividad mínimo.

Cómo escalar verticalmente un clúster de Valkey o Redis OSS de un solo nodo (API de ElastiCache)

1. Determine los tipos de nodos a los que puede escalar verticalmente al ejecutar la acción `ListAllowedNodeTypeModifications` de la API de ElastiCache con el siguiente parámetro.

   - `CacheClusterId`: nombre del clúster de Valkey o Redis OSS de un solo nodo que desea escalar verticalmente.

   ```
   https://elasticache.us-west-2.amazonaws.com/
       ?Action=ListAllowedNodeTypeModifications
       &CacheClusterId=MyRedisCacheCluster
       &Version=2015-02-02
       &SignatureVersion=4
       &SignatureMethod=HmacSHA256
       &Timestamp=20150202T192317Z
       &X-Amz-Credential=<credential>
   ```

   Para obtener más información, consulte ListAllowedNodeTypeModifications en la Referencia de la API de Amazon ElastiCache.

2. Modifique el clúster existente especificando el clúster que se debe escalar verticalmente e indique el nuevo tipo de nodo más grande con la API `ModifyCacheCluster` de ElastiCache y los siguientes parámetros.

   - `CacheClusterId`: nombre del clúster que está escalando verticalmente.

- `CacheNodeType`: tipo de nodo nuevo, de mayor tamaño, al que desea escalar verticalmente el clúster. Este valor debe ser uno de los tipos de nodos devueltos por la acción `ListAllowedNodeTypeModifications` en el paso anterior.
- `CacheParameterGroupName`: [Opcional] Utilice este parámetro si va a utilizar `reserved-memory` para administrar la memoria reservada de su clúster. Especifique un grupo de parámetros de caché personalizado que reserve la cantidad correcta de memoria para el nuevo tipo de nodo. Si va a utilizar `reserved-memory-percent`, puede omitir este parámetro.
- `ApplyImmediately`: establézcalo en `true` para que el proceso de escalado vertical se realice de inmediato. Para aplazar el proceso de ampliación al siguiente periodo de mantenimiento del clúster, utilice `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=ModifyCacheCluster
    &ApplyImmediately=true
    &CacheClusterId=MyRedisCacheCluster
    &CacheNodeType=cache.m3.xlarge
    &CacheParameterGroupName redis32-m2-xl
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
    &X-Amz-Credential=<credential>
```

Para obtener más información, consulte ModifyCacheCluster en la Referencia de la API de Amazon ElastiCache.

3. Si ha utilizado `ApplyImmediately=true`, compruebe el estado del nuevo clúster con la API `DescribeCacheClusters` de ElastiCache con el siguiente parámetro. Cuando el estado cambie a disponible, podrá comenzar a usar el nuevo clúster de mayor tamaño.

   - `CacheClusterId`: nombre del clúster de Valkey o Redis OSS de un solo nodo. Use este parámetro para describir un clúster determinado en lugar de todos los clústeres.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=DescribeCacheClusters
    &CacheClusterId=MyRedisCacheCluster
    &Version=2015-02-02
```

```
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
    &X-Amz-Credential=<credential>
```

Para obtener más información, consulte DescribeCacheClusters en la Referencia de la API de Amazon ElastiCache.

Reducción vertical de clústeres de Valkey o Redis OSS de un solo nodo

En las secciones siguientes, se muestra cómo reducir verticalmente un clúster de Valkey o Redis OSS de un solo nodo a un tipo de nodo más pequeño. Para el éxito a largo plazo de su nuevo clúster de Valkey o Redis OSS, es importante asegurarse de que el nuevo tipo de nodo más pequeño sea lo suficientemente grande para alojar todos los datos, además de asumir la sobrecarga de Valkey o Redis OSS. Para obtener más información, consulte Forma de garantizar que dispone de memoria suficiente para crear una instantánea de Valkey o Redis OSS.

> **Note**
>
> Para los clústeres que ejecutan el tipo de nodo r6gd, solo puede escalar a tamaños de nodo dentro de la familia de nodos r6gd.

Temas

- Reducción vertical de clústeres de Valkey o Redis OSS de un solo nodo (consola)
- Reducción vertical de clústeres de Valkey o Redis OSS de un solo nodo (AWS CLI)
- Reducción vertical de clústeres de Valkey o Redis OSS de un solo nodo (API de ElastiCache)

Reducción vertical de clústeres de Valkey o Redis OSS de un solo nodo (consola)

En el siguiente procedimiento, se le mostrará cómo reducir un clúster de Valkey o Redis OSS de un solo nodo a un tipo de nodo más pequeño mediante la consola de ElastiCache.

> **Important**
>
> Si su grupo de parámetros usa `reserved-memory` para apartar memoria para la capacidad adicional de Valkey o Redis OSS, antes de comenzar el escalado, asegúrese de tener un grupo de parámetros personalizado que reserve la cantidad de memoria adecuada para el nuevo tipo de nodo. De forma alternativa, puede modificar un grupo de parámetros personalizado para que use `reserved-memory-percent` y utilizar ese grupo de parámetros para su nuevo clúster.
> Si va a utilizar `reserved-memory-percent`, esto no es necesario.
> Para obtener más información, consulte Administración de la memoria reservada para Valkey y Redis OSS.

Reducción vertical de un clúster de Valkey o Redis OSS de un solo nodo (consola)

1. Asegúrese de que el tipo de nodo más pequeño es adecuado para sus necesidades de datos y de capacidad adicional.
2. Si su grupo de parámetros usa `reserved-memory` para apartar memoria para la sobrecarga de Valkey o Redis OSS, asegúrese de tener un grupo de parámetros personalizado que aparte la cantidad de memoria adecuada para el nuevo tipo de nodo.

   De forma alternativa, puede modificar su grupo de parámetros personalizado para utilizar `reserved-memory-percent`. Para obtener más información, consulte Administración de la memoria reservada para Valkey y Redis OSS.
3. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
4. En la lista de clústeres, seleccione el clúster que desee reducir. Este clúster debe ejecutar el motor de Valkey o Redis OSS y no el motor de Valkey o Redis OSS en clúster.
5. Elija Modificar.
6. En el asistente Modify Cluster:

   a. Elija el tipo de nodo que desee reducir en la lista Node type (Tipo de nodo).

   b. Si va a utilizar `reserved-memory` para administrar su memoria, en la lista Parameter Group, elija el grupo de parámetros personalizado que reserve la cantidad correcta de memoria para su nuevo tipo de nodo.
7. Si desea realizar el proceso de reducción de forma inmediata, elija la casilla de verificación Apply immediately (Aplicar inmediatamente). Si no se deja seleccionada la casilla de verificación Apply immediately (Aplicar inmediatamente), el proceso de reducción se realizará durante el siguiente periodo de mantenimiento de este clúster.
8. Elija Modificar.
9. Cuando el estado del clúster cambie de modifying a available, el clúster se habrá escalado al nuevo tipo de nodo. No es necesario actualizar los puntos de enlace de su aplicación.

Reducción vertical de clústeres de Valkey o Redis OSS de un solo nodo (AWS CLI)

El siguiente procedimiento describe cómo reducir verticalmente un clúster de Valkey o Redis OSS de un solo nodo mediante la AWS CLI.

Cómo reducir verticalmente un clúster de Valkey o Redis OSS de un solo nodo (AWS CLI)

1. Determine los tipos de nodos a los que puede aplicar el escalado descendente ejecutando el comando AWS CLI de la `list-allowed-node-type-modifications` con el siguiente parámetro.

   - `--cache-cluster-id`

   Para Linux, macOS o Unix:

   ```
   aws elasticache list-allowed-node-type-modifications \
        --cache-cluster-id my-cache-cluster-id
   ```

   Para Windows:

   ```
   aws elasticache list-allowed-node-type-modifications ^
        --cache-cluster-id my-cache-cluster-id
   ```

   La salida del comando anterior es similar a la siguiente (formato JSON).

   ```
   {
        "ScaleUpModifications": [
            "cache.m3.2xlarge",
            "cache.m3.large",
            "cache.m3.xlarge",
            "cache.m4.10xlarge",
            "cache.m4.2xlarge",
            "cache.m4.4xlarge",
            "cache.m4.large",
            "cache.m4.xlarge",
            "cache.r3.2xlarge",
            "cache.r3.4xlarge",
            "cache.r3.8xlarge",
            "cache.r3.large",
            "cache.r3.xlarge"
        ]
           "ScaleDownModifications": [
            "cache.t2.micro",
            "cache.t2.small ",
            "cache.t2.medium ",
               "cache.t1.small ",
   ```

```
    ],

}
```

Para obtener más información consulte list-allowed-node-type-modifications en la Referencia de la AWS CLI.

2. Modifique el clúster existente especificando el clúster que se debe reducir verticalmente e indique el nuevo tipo de nodo más pequeño con el comando `modify-cache-cluster` de la AWS CLI y los siguientes parámetros.

   - `--cache-cluster-id`: nombre del clúster que está reduciendo verticalmente.
   - `--cache-node-type`: tipo de nodo nuevo al que desea escalar el clúster. Este valor debe ser uno de los tipos de nodos devueltos por el comando `list-allowed-node-type-modifications` en el paso 1.
   - `--cache-parameter-group-name`: [Opcional] Utilice este parámetro si va a utilizar `reserved-memory` para administrar la memoria reservada de su clúster. Especifique un grupo de parámetros de caché personalizado que reserve la cantidad correcta de memoria para el nuevo tipo de nodo. Si va a utilizar `reserved-memory-percent`, puede omitir este parámetro.
   - `--apply-immediately`: hace que el proceso de reducción vertical se aplique de inmediato. Para aplazar el proceso de ampliación al siguiente periodo de mantenimiento del clúster, utilice el parámetro `--no-apply-immediately`.

   Para Linux, macOS o Unix:

   ```
   aws elasticache modify-cache-cluster \
        --cache-cluster-id my-redis-cache-cluster \
        --cache-node-type cache.m3.xlarge \
        --cache-parameter-group-name redis32-m2-xl \
        --apply-immediately
   ```

   Para Windows:

   ```
   aws elasticache modify-cache-cluster ^
        --cache-cluster-id my-redis-cache-cluster ^
        --cache-node-type cache.m3.xlarge ^
        --cache-parameter-group-name redis32-m2-xl ^
   ```

```
    --apply-immediately
```

La salida del comando anterior es similar a la siguiente (formato JSON).

```
{
    "CacheCluster": {
        "Engine": "redis",
        "CacheParameterGroup": {
            "CacheNodeIdsToReboot": [],
            "CacheParameterGroupName": "default.redis6,x",
            "ParameterApplyStatus": "in-sync"
        },
        "SnapshotRetentionLimit": 1,
        "CacheClusterId": "my-redis-cache-cluster",
        "CacheSecurityGroups": [],
        "NumCacheNodes": 1,
        "SnapshotWindow": "00:00-01:00",
        "CacheClusterCreateTime": "2017-02-21T22:34:09.645Z",
        "AutoMinorVersionUpgrade": true,
        "CacheClusterStatus": "modifying",
        "PreferredAvailabilityZone": "us-west-2a",
        "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
        "CacheSubnetGroupName": "default",
        "EngineVersion": "6.0",
        "PendingModifiedValues": {
            "CacheNodeType": "cache.m3.2xlarge"
        },
        "PreferredMaintenanceWindow": "tue:11:30-tue:12:30",
        "CacheNodeType": "cache.m3.medium",
         "DataTiering": "disabled"
    }
 }
```

Para obtener más información consulte modify-cache-cluster en la Referencia de la AWS CLI.

3. Si ha utilizado `--apply-immediately`, compruebe el estado del nuevo clúster con el comando `describe-cache-clusters` de la AWS CLI con el siguiente parámetro. Cuando el estado cambie a disponible, podrá comenzar a usar el nuevo clúster de mayor tamaño.

   - `--cache-cluster-id`: nombre del clúster de Valkey o Redis OSS de un solo nodo. Use este parámetro para describir un clúster determinado en lugar de todos los clústeres.

```
aws elasticache describe-cache-clusters --cache-cluster-id my-redis-cache-cluster
```

Para obtener más información consulte describe-cache-clusters en la Referencia de la AWS CLI.

Reducción vertical de clústeres de Valkey o Redis OSS de un solo nodo (API de ElastiCache)

El siguiente procedimiento describe cómo reducir verticalmente un clúster de Valkey o Redis OSS de un solo nodo mediante la API de ElastiCache.

Cómo reducir verticalmente un clúster de Valkey o Redis OSS de un solo nodo (API de ElastiCache)

1. Determine los tipos de nodos a los que puede reducir verticalmente al ejecutar la acción `ListAllowedNodeTypeModifications` de la API de ElastiCache con el siguiente parámetro.

   - `CacheClusterId`: nombre del clúster de Valkey o Redis OSS de un solo nodo que desea reducir verticalmente.

   ```
   https://elasticache.us-west-2.amazonaws.com/
       ?Action=ListAllowedNodeTypeModifications
       &CacheClusterId=MyRedisCacheCluster
       &Version=2015-02-02
       &SignatureVersion=4
       &SignatureMethod=HmacSHA256
       &Timestamp=20150202T192317Z
       &X-Amz-Credential=<credential>
   ```

   Para obtener más información, consulte ListAllowedNodeTypeModifications en la Referencia de la API de Amazon ElastiCache.

2. Modifique el clúster existente especificando el clúster que se debe escalar verticalmente e indique el nuevo tipo de nodo más grande con la API `ModifyCacheCluster` de ElastiCache y los siguientes parámetros.

   - `CacheClusterId`: nombre del clúster que está reduciendo verticalmente.
   - `CacheNodeType`: tipo de nodo nuevo, de mayor tamaño, al que desea reducir verticalmente el clúster. Este valor debe ser uno de los tipos de nodos devueltos por la acción `ListAllowedNodeTypeModifications` en el paso anterior.

- `CacheParameterGroupName`: [Opcional] Utilice este parámetro si va a utilizar `reserved-memory` para administrar la memoria reservada de su clúster. Especifique un grupo de parámetros de caché personalizado que reserve la cantidad correcta de memoria para el nuevo tipo de nodo. Si va a utilizar `reserved-memory-percent`, puede omitir este parámetro.
- `ApplyImmediately`: establézcalo en `true` para que el proceso de reducción vertical se realice de inmediato. Para aplazar el proceso de ampliación al siguiente periodo de mantenimiento del clúster, utilice `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=ModifyCacheCluster
    &ApplyImmediately=true
    &CacheClusterId=MyRedisCacheCluster
    &CacheNodeType=cache.m3.xlarge
    &CacheParameterGroupName redis32-m2-xl
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
    &X-Amz-Credential=<credential>
```

Para obtener más información, consulte ModifyCacheCluster en la Referencia de la API de Amazon ElastiCache.

3. Si ha utilizado `ApplyImmediately=true`, compruebe el estado del nuevo clúster con la API `DescribeCacheClusters` de ElastiCache con el siguiente parámetro. Cuando el estado cambie a disponible, puede comenzar con el nuevo clúster de menor tamaño.

   - `CacheClusterId`: nombre del clúster de Valkey o Redis OSS de un solo nodo. Use este parámetro para describir un clúster determinado en lugar de todos los clústeres.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=DescribeCacheClusters
    &CacheClusterId=MyRedisCacheCluster
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
```

```
    &X-Amz-Credential=<credential>
```

Para obtener más información, consulte DescribeCacheClusters en la Referencia de la API de Amazon ElastiCache.

Escalado de nodos de réplica para Valkey o Redis OSS (modo de clúster deshabilitado)

Un clúster de Valkey o Redis OSS con nodos de réplica (lo que se conoce como grupo de replicación en la API/CLI) proporciona un alto nivel de disponibilidad a través de la replicación con multi-AZ y la conmutación por error automática habilitadas. Un clúster con nodos de réplica es una recopilación lógica de hasta seis nodos de Valkey o Redis OSS en los que un nodo, el principal, puede atender solicitudes tanto de lectura como de escritura. Todos los demás nodos del clúster son réplicas de solo lectura del principal. Los datos que se escriben en el principal se replican de forma asíncrona en todas las réplicas de lectura del clúster. Puesto que Valkey o Redis OSS (modo de clúster deshabilitado) no admite la partición de los datos entre varios clústeres, cada nodo de un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) contiene el conjunto de datos de caché completo. Los clústeres de Valkey o Redis OSS (modo de clúster habilitado) admiten la partición de los datos en hasta 500 particiones.

Para cambiar la capacidad de datos de su clúster, debe ampliarlo a un tipo de nodo más grande, o reducirlo a un tipo de nodo más pequeño.

Para cambiar la capacidad de lectura del clúster, agregue más réplicas de lectura, hasta un máximo de 5, o bien elimine réplicas de lectura.

El proceso de escalado vertical de ElastiCache se ha diseñado para hacer todo lo posible por retener sus datos y necesita que la replicación de Valkey o Redis OSS se realice correctamente. Para clústeres de Valkey o Redis OSS con réplicas, recomendamos disponer de suficiente memoria para Valkey o Redis OSS.

Temas

- Escalado vertical de clústeres de Valkey o Redis OSS con réplicas
- Reducción vertical de clústeres de Valkey o Redis OSS con réplicas
- Aumento de la capacidad de lectura
- Reducción de la capacidad de lectura

Temas relacionados

- Alta disponibilidad a través de grupos de reproducción
- Replicación: Valkey y Redis OSS en modo de clúster deshabilitado o habilitado
- Minimización del tiempo de inactividad en ElastiCache utilizando multi-AZ con Valkey y Redis OSS

- Forma de garantizar que dispone de memoria suficiente para crear una instantánea de Valkey o Redis OSS

Temas

- Escalado vertical de clústeres de Valkey o Redis OSS con réplicas
- Reducción vertical de clústeres de Valkey o Redis OSS con réplicas
- Aumento de la capacidad de lectura
- Reducción de la capacidad de lectura

Escalado vertical de clústeres de Valkey o Redis OSS con réplicas

Amazon ElastiCache proporciona compatibilidad con la consola, la CLI y la API para escalar verticalmente su grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado).

Al iniciar el proceso de escalado vertical, ElastiCache realiza lo siguiente:

1. Lanza un grupo de reproducción con el nuevo tipo de nodo.
2. Copia todos los datos del nodo principal actual al nuevo nodo principal.
3. Sincroniza las nuevas réplicas de lectura con el nuevo nodo principal.
4. Actualiza los entradas de DNS para que apunten a los nuevos nodos. Por ello, no tiene que actualizar los puntos de enlace de su aplicación. En Valkey 7.2 y versiones posteriores o en Redis OSS 5.0.5 y versiones posteriores, puede escalar clústeres con la conmutación por error habilitada mientras el clúster permanece en línea y atiende solicitudes de entrada. En Redis OSS 4.0.10 y versiones anteriores, puede observar una breve interrupción de las lecturas y escrituras del nodo principal mientras se actualiza la entrada de DNS.
5. Elimina los nodos antiguos (CLI/API: grupo de reproducción). Observará una breve interrupción (unos segundos) de las operaciones de lectura y escritura de los nodos anteriores porque se desconectarán las conexiones con los nodos anteriores.

El tiempo que tarda este proceso depende de su tipo de nodo y de la cantidad de datos que haya en su clúster.

Tal y como se muestra en la siguiente tabla, la operación de escalado vertical de Valkey o Redis OSS se bloqueará si tiene una actualización de motor programada para el próximo periodo de mantenimiento del clúster.

Operaciones bloqueadas de Valkey o Redis OSS

Operaciones pendientes	Operaciones bloqueadas
Escalado ascendente	Actualización del motor inmediata
Actualización del motor	Ampliación inmediata
Ampliación y actualización del motor	Ampliación inmediata
	Actualización del motor inmediata

Si hay una operación pendiente que esté bloqueando, tiene las siguientes opciones.

- Programe la operación de escalado vertical de Valkey o Redis OSS para el siguiente periodo de mantenimiento desactivando la casilla de verificación Aplicar inmediatamente (en la CLI: use `--no-apply-immediately`, en la API: use `ApplyImmediately=false`).
- Espere al siguiente periodo de mantenimiento (o después) para realizar la operación de escalado vertical de Valkey o Redis OSS.
- Añada la actualización del motor de Valkey o Redis OSS a esta modificación de clúster con la casilla de verificación Aplicar inmediatamente activada (en la CLI, use `--apply-immediately` y en la API, `ApplyImmediately=true`). Esto desbloqueará su operación de ampliación, ya que hará que la actualización del motor se realice inmediatamente.

En las siguientes secciones, se describe cómo escalar verticalmente el clúster de Valkey o Redis OSS con réplicas mediante la consola de ElastiCache, la AWS CLI y la API de ElastiCache.

> **Important**
>
> Si su grupo de parámetros usa `reserved-memory` para apartar memoria para la capacidad adicional de Valkey o Redis OSS, antes de comenzar el escalado, asegúrese de tener un grupo de parámetros personalizado que reserve la cantidad de memoria adecuada para el nuevo tipo de nodo. De forma alternativa, puede modificar un grupo de parámetros personalizado para que use `reserved-memory-percent` y utilizar ese grupo de parámetros para su nuevo clúster.
> Si va a utilizar `reserved-memory-percent`, esto no es necesario.
> Para obtener más información, consulte Administración de la memoria reservada para Valkey y Redis OSS.

Escalado vertical de clústeres de Valkey o Redis OSS con réplicas (consola)

El tiempo que se tarda en la ampliación a un tipo de nodo más grande varía en función del tipo de nodo y de la cantidad de datos de su clúster actual.

El siguiente proceso escala su clúster con réplicas de su tipo de nodo actual a un nuevo tipo de nodo más grande mediante la consola de ElastiCache. Durante este proceso, puede haber una breve interrupción en las operaciones de lectura y escritura de otras versiones en el nodo principal mientras se actualiza la entrada de DNS. Es posible que observe un tiempo de inactividad inferior

a 1 segundo en los nodos que ejecutan la versión 5.0.6 y posteriores, y de varios segundos en versiones anteriores.

Escalado vertical de clústeres de Valkey o Redis OSS con réplicas (consola)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Clústeres de Valkey o Clústeres de Redis OSS.
3. En la lista de clústeres, seleccione el clúster que desea ampliar. Este clúster debe ejecutar el motor de Valkey o Redis OSS y no el motor de Valkey o Redis OSS en clúster.
4. Elija Modificar.
5. En el asistente Modify Cluster:
    a. Elija el tipo de nodo que desee ampliar en la lista Node type. Tenga en cuenta que no todos los tipos de nodos están disponibles para el proceso de reducción.
    b. Si va a utilizar `reserved-memory` para administrar su memoria, en la lista Parameter Group, elija el grupo de parámetros personalizado que reserve la cantidad correcta de memoria para su nuevo tipo de nodo.
6. Si desea realizar el proceso de ampliación de forma inmediata, elija la casilla de verificación Apply immediately. Si no se deja seleccionada la casilla de verificación Apply immediately, el proceso de ampliación se realizará durante el siguiente periodo de mantenimiento de este clúster.
7. Elija Modificar.
8. Cuando el estado del clúster cambie de modifying a available, el clúster se habrá escalado al nuevo tipo de nodo. No es necesario actualizar los puntos de enlace de su aplicación.

Escalado vertical de un grupo de replicación de Valkey o Redis OSS (AWS CLI)

El siguiente proceso amplía su grupo de reproducción de su tipo de nodo actual a un nuevo tipo de nodo más grande utilizando AWS CLI. Durante este proceso, ElastiCache actualiza las entradas de DNS para que apunten a los nuevos nodos. Por ello, no tiene que actualizar los puntos de enlace de su aplicación. En Valkey 7.2 y versiones posteriores o en Redis OSS 5.0.5 y versiones posteriores, puede escalar clústeres con la conmutación por error habilitada mientras el clúster permanece en línea y atiende solicitudes de entrada. En la versión 4.0.10 y anteriores, puede observar una breve interrupción de las versiones de lectura y escritura del nodo principal mientras se actualiza la entrada de DNS.

El tiempo que se tarda en el escalado vertical a un tipo de nodo más grande varía en función de su tipo de nodo y de la cantidad de datos de su clúster actual.

Escalado vertical de un grupo de replicación de Valkey o Redis OSS (AWS CLI)

1. Determine los tipos de nodos a los que puede ampliar ejecutando el comando AWS CLI de la `list-allowed-node-type-modifications` con el siguiente parámetro.

   - `--replication-group-id`: nombre del grupo de replicación. Use este parámetro para describir un determinado grupo de replicación en lugar de todos los grupos de replicación.

   Para Linux, macOS o Unix:

   ```
   aws elasticache list-allowed-node-type-modifications \
        --replication-group-id my-repl-group
   ```

   Para Windows:

   ```
   aws elasticache list-allowed-node-type-modifications ^
        --replication-group-id my-repl-group
   ```

   La salida de esta operación tiene un aspecto similar al siguiente (formato JSON).

   ```
   {
        "ScaleUpModifications": [
            "cache.m3.2xlarge",
            "cache.m3.large",
            "cache.m3.xlarge",
            "cache.m4.10xlarge",
            "cache.m4.2xlarge",
            "cache.m4.4xlarge",
            "cache.m4.large",
            "cache.m4.xlarge",
            "cache.r3.2xlarge",
            "cache.r3.4xlarge",
            "cache.r3.8xlarge",
            "cache.r3.large",
            "cache.r3.xlarge"
        ]
    }
   ```

Para obtener más información consulte list-allowed-node-type-modifications en la Referencia de la AWS CLI.

2. Amplíe su grupo de replicación actual al nuevo tipo de nodo utilizando el comando AWS CLI de la `modify-replication-group` con los siguientes parámetros.

   - `--replication-group-id`: nombre del grupo de replicación.
   - `--cache-node-type`: nuevo tipo de nodo más grande de los clústeres en este grupo de replicación. Este valor debe ser uno de los tipos de instancia devueltos por el comando `list-allowed-node-type-modifications` en el paso anterior.
   - `--cache-parameter-group-name`: [Opcional] Utilice este parámetro si va a utilizar `reserved-memory` para administrar la memoria reservada de su clúster. Especifique un grupo de parámetros de caché personalizado que reserve la cantidad correcta de memoria para el nuevo tipo de nodo. Si va a utilizar `reserved-memory-percent`, puede omitir este parámetro.
   - `--apply-immediately`: hace que el proceso de escalado vertical se aplique de inmediato. Para aplazar la operación de ampliación al siguiente periodo de mantenimiento, utilice `--no-apply-immediately`.

   Para Linux, macOS o Unix:

   ```
   aws elasticache modify-replication-group \
        --replication-group-id my-repl-group \
        --cache-node-type cache.m3.xlarge \
        --cache-parameter-group-name redis32-m3-2xl \
        --apply-immediately
   ```

   Para Windows:

   ```
   aws elasticache modify-replication-group ^
        --replication-group-id my-repl-group ^
        --cache-node-type cache.m3.xlarge ^
        --cache-parameter-group-name redis32-m3-2xl \
        --apply-immediately
   ```

   La salida de este comando tiene un aspecto similar al siguiente (formato JSON).

```
{
 "ReplicationGroup": {
  "Status": "available",
  "Description": "Some description",
  "NodeGroups": [{
   "Status": "available",
   "NodeGroupMembers": [{
     "CurrentRole": "primary",
     "PreferredAvailabilityZone": "us-west-2b",
     "CacheNodeId": "0001",
     "ReadEndpoint": {
      "Port": 6379,
      "Address": "my-repl-group-001.8fdx4s.0001.usw2.cache.amazonaws.com"
     },
     "CacheClusterId": "my-repl-group-001"
    },
    {
     "CurrentRole": "replica",
     "PreferredAvailabilityZone": "us-west-2c",
     "CacheNodeId": "0001",
     "ReadEndpoint": {
      "Port": 6379,
      "Address": "my-repl-group-002.8fdx4s.0001.usw2.cache.amazonaws.com"
     },
     "CacheClusterId": "my-repl-group-002"
    }
   ],
   "NodeGroupId": "0001",
   "PrimaryEndpoint": {
    "Port": 6379,
    "Address": "my-repl-group.8fdx4s.ng.0001.usw2.cache.amazonaws.com"
   }
  }],
  "ReplicationGroupId": "my-repl-group",
  "SnapshotRetentionLimit": 1,
  "AutomaticFailover": "disabled",
  "SnapshotWindow": "12:00-13:00",
  "SnapshottingClusterId": "my-repl-group-002",
  "MemberClusters": [
   "my-repl-group-001",
   "my-repl-group-002"
  ],
```

```
  "PendingModifiedValues": {}
 }
}
```

Para obtener más información consulte modify-replication-group en la Referencia de la AWS CLI.

3. Si utilizó el parámetro `--apply-immediately`, monitorice el estado del grupo de reproducción utilizando el comando AWS CLI de la `describe-replication-group` con el siguiente parámetro. Mientras el estado sea en modificación, es posible que observe un tiempo de inactividad de un segundo en los nodos que ejecutan la versión 5.0.6 y posteriores y una breve interrupción de las operaciones de lectura y escritura con las versiones anteriores en el nodo principal mientras se actualiza la entrada de DNS.

   - `--replication-group-id`: nombre del grupo de replicación. Use este parámetro para describir un determinado grupo de replicación en lugar de todos los grupos de replicación.

   Para Linux, macOS o Unix:

   ```
   aws elasticache describe-replication-groups \
        --replication-group-id my-replication-group
   ```

   Para Windows:

   ```
   aws elasticache describe-replication-groups ^
        --replication-group-id my-replication-group
   ```

   Para obtener más información, consulte describe-replication-groups en la Referencia de la AWS CLI.

Escalado vertical de un grupo de replicación de Valkey o Redis OSS (API de ElastiCache)

El siguiente proceso escala el grupo de reproducción de su tipo de nodo actual a un nuevo tipo de nodo más grande mediante la API de ElastiCache. En Valkey 7.2 y versiones posteriores o en Redis OSS 5.0.5 y versiones posteriores, puede escalar clústeres con la conmutación por error habilitada mientras el clúster permanece en línea y atiende solicitudes de entrada. En Redis OSS 4.0.10 y versiones anteriores, puede observar una breve interrupción de las lecturas y escrituras del nodo principal mientras se actualiza la entrada de DNS.

El tiempo que se tarda en el escalado vertical a un tipo de nodo más grande varía en función de su tipo de nodo y de la cantidad de datos de su clúster actual.

Escalado vertical de un grupo de replicación de Valkey o Redis OSS (API de ElastiCache)

1. Determine a qué tipos de nodos puede escalar verticalmente mediante la acción `ListAllowedNodeTypeModifications` de la API de ElastiCache con el siguiente parámetro.

   - `ReplicationGroupId`: nombre del grupo de replicación. Use este parámetro para describir un grupo de replicación específico en lugar de todos los grupos de replicación.

   ```
   https://elasticache.us-west-2.amazonaws.com/
       ?Action=ListAllowedNodeTypeModifications
       &ReplicationGroupId=MyReplGroup
       &Version=2015-02-02
       &SignatureVersion=4
       &SignatureMethod=HmacSHA256
       &Timestamp=20150202T192317Z
       &X-Amz-Credential=<credential>
   ```

   Para obtener más información, consulte ListAllowedNodeTypeModifications en la Referencia de la API de Amazon ElastiCache.

2. Escale verticalmente su grupo de reproducción actual al nuevo tipo de nodo mediante la acción `ModifyReplicationGroup` de la API de ElastiCache y con los siguientes parámetros.

   - `ReplicationGroupId`: nombre del grupo de replicación.
   - `CacheNodeType`: nuevo tipo de nodo más grande de los clústeres en este grupo de replicación. Este valor debe ser uno de los tipos de instancia devueltos por la acción `ListAllowedNodeTypeModifications` en el paso anterior.
   - `CacheParameterGroupName`: [Opcional] Utilice este parámetro si va a utilizar `reserved-memory` para administrar la memoria reservada de su clúster. Especifique un grupo de parámetros de caché personalizado que reserve la cantidad correcta de memoria para el nuevo tipo de nodo. Si va a utilizar `reserved-memory-percent`, puede omitir este parámetro.
   - `ApplyImmediately`: establézcalo en `true` para que el proceso de escalado vertical se aplique de inmediato. Para aplazar el proceso de ampliación al siguiente periodo de mantenimiento, utilice `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=ModifyReplicationGroup
    &ApplyImmediately=true
    &CacheNodeType=cache.m3.2xlarge
    &CacheParameterGroupName=redis32-m3-2xl
    &ReplicationGroupId=myReplGroup
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &Version=2014-12-01
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20141201T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Para obtener más información, consulte ModifyReplicationGroup en la Referencia de la API de Amazon ElastiCache.

3. Si utilizó `ApplyImmediately=true`, monitoree el estado del grupo de reproducción mediante la acción `DescribeReplicationGroups` de la API de ElastiCache con los siguientes parámetros. Cuando el estado cambie de modifying a available, podrá empezar a escribir en su nuevo grupo de replicación ampliado.

   - `ReplicationGroupId`: nombre del grupo de replicación. Use este parámetro para describir un determinado grupo de replicación en lugar de todos los grupos de replicación.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=DescribeReplicationGroups
    &ReplicationGroupId=MyReplGroup
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
    &X-Amz-Credential=<credential>
```

Para obtener más información, consulte DescribeReplicationGroups en la Referencia de la API de Amazon ElastiCache.

Reducción vertical de clústeres de Valkey o Redis OSS con réplicas

En las secciones siguientes, se muestra cómo reducir verticalmente un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) con nodos de réplica a un tipo de nodo más pequeño. Para que el proceso se realice correctamente, es muy importante asegurarse de que el nuevo tipo de nodo más pequeño sea lo suficientemente grande para alojar todos los datos, más la capacidad adicional. Para obtener más información, consulte Forma de garantizar que dispone de memoria suficiente para crear una instantánea de Valkey o Redis OSS.

> **Note**
>
> Para los clústeres que ejecutan el tipo de nodo r6gd, solo puede escalar a tamaños de nodo dentro de la familia de nodos r6gd.

> **Important**
>
> Si su grupo de parámetros usa `reserved-memory` para apartar memoria para la capacidad adicional de Valkey o Redis OSS, antes de comenzar el escalado, asegúrese de tener un grupo de parámetros personalizado que reserve la cantidad de memoria adecuada para el nuevo tipo de nodo. De forma alternativa, puede modificar un grupo de parámetros personalizado para que use `reserved-memory-percent` y utilizar ese grupo de parámetros para su nuevo clúster.
> Si va a utilizar `reserved-memory-percent`, esto no es necesario.
> Para obtener más información, consulte Administración de la memoria reservada para Valkey y Redis OSS.

Reducción vertical de clústeres de Valkey o Redis OSS con réplicas (consola)

El siguiente proceso escala su clúster de Valkey o Redis OSS con nodos de réplica a un tipo de nodo más pequeño mediante la consola de ElastiCache.

Reducción vertical de un clúster de Valkey o Redis OSS con nodos de réplica (consola)

1. Asegúrese de que el tipo de nodo más pequeño es adecuado para sus necesidades de datos y de capacidad adicional.

2. Si su grupo de parámetros usa `reserved-memory` para apartar memoria para la sobrecarga de Valkey o Redis OSS, asegúrese de tener un grupo de parámetros personalizado que aparte la cantidad de memoria adecuada para el nuevo tipo de nodo.

   De forma alternativa, puede modificar su grupo de parámetros personalizado para utilizar `reserved-memory-percent`. Para obtener más información, consulte Administración de la memoria reservada para Valkey y Redis OSS.
3. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
4. En la lista de clústeres, seleccione el clúster que desee reducir. Este clúster debe ejecutar el motor de Valkey o Redis OSS y no el motor de Valkey o Redis OSS en clúster.
5. Elija Modificar.
6. En el asistente Modify Cluster:

   a. Elija el tipo de nodo que desee reducir en la lista Node type (Tipo de nodo).

   b. Si va a utilizar `reserved-memory` para administrar su memoria, en la lista Parameter Group, elija el grupo de parámetros personalizado que reserve la cantidad correcta de memoria para su nuevo tipo de nodo.
7. Si desea realizar el proceso de reducción de forma inmediata, elija la casilla de verificación Apply immediately (Aplicar inmediatamente). Si no se deja seleccionada la casilla de verificación Apply immediately (Aplicar inmediatamente), el proceso de reducción se realizará durante el siguiente periodo de mantenimiento de este clúster.
8. Elija Modificar.
9. Cuando el estado del clúster cambie de modifying a available, el clúster se habrá escalado al nuevo tipo de nodo. No es necesario actualizar los puntos de enlace de su aplicación.

Reducción vertical de un grupo de replicación de Valkey o Redis OSS (AWS CLI)

El siguiente proceso escala su grupo de reproducción de su tipo de nodo actual a un nuevo tipo de nodo más pequeño utilizando AWS CLI. Durante este proceso, ElastiCache actualiza las entradas de DNS para que apunten a los nuevos nodos. Por ello, no tiene que actualizar los puntos de enlace de su aplicación. En Valkey 7.2 y versiones posteriores o en Redis OSS 5.0.5 y versiones posteriores, puede escalar clústeres con la conmutación por error habilitada mientras el clúster permanece en línea y atiende solicitudes de entrada. En la versión 4.0.10 y anteriores, puede observar una breve interrupción de las versiones de lectura y escritura del nodo principal mientras se actualiza la entrada de DNS.

Sin embargo, las lecturas de los clústeres de réplica de lectura continuarán de forma ininterrumpida.

El tiempo que se tarda en la realización del escalado descendente a un tipo de nodo más pequeño varía en función de su tipo de nodo y de la cantidad de datos de su clúster actual.

Reducción vertical de un grupo de replicación de Valkey o Redis OSS (AWS CLI)

1. Determine los tipos de nodos a los que puede reducir ejecutando el comando AWS CLI de la `list-allowed-node-type-modifications` con el siguiente parámetro.

   - `--replication-group-id`: nombre del grupo de replicación. Use este parámetro para describir un determinado grupo de replicación en lugar de todos los grupos de replicación.

   Para Linux, macOS o Unix:

   ```
   aws elasticache list-allowed-node-type-modifications \
        --replication-group-id my-repl-group
   ```

   Para Windows:

   ```
   aws elasticache list-allowed-node-type-modifications ^
        --replication-group-id my-repl-group
   ```

   La salida de esta operación tiene un aspecto similar al siguiente (formato JSON).

   ```
   {
        "ScaleDownModifications": [
            "cache.m3.2xlarge",
            "cache.m3.large",
            "cache.m3.xlarge",
            "cache.m4.10xlarge",
            "cache.m4.2xlarge",
            "cache.m4.4xlarge",
            "cache.m4.large",
            "cache.m4.xlarge",
            "cache.r3.2xlarge",
            "cache.r3.4xlarge",
            "cache.r3.8xlarge",
            "cache.r3.large",
            "cache.r3.xlarge"
        ]
   ```

```
}
```

Para obtener más información consulte list-allowed-node-type-modifications en la Referencia de la AWS CLI.

2. Amplíe su grupo de replicación actual al nuevo tipo de nodo utilizando el comando AWS CLI de la `modify-replication-group` con los siguientes parámetros.

   - `--replication-group-id`: nombre del grupo de replicación.
   - `--cache-node-type`: nuevo tipo de nodo más pequeño de los clústeres en este grupo de replicación. Este valor debe ser uno de los tipos de instancia devueltos por el comando `list-allowed-node-type-modifications` en el paso anterior.
   - `--cache-parameter-group-name`: [Opcional] Utilice este parámetro si va a utilizar `reserved-memory` para administrar la memoria reservada de su clúster. Especifique un grupo de parámetros de caché personalizado que reserve la cantidad correcta de memoria para el nuevo tipo de nodo. Si va a utilizar `reserved-memory-percent`, puede omitir este parámetro.
   - `--apply-immediately`: hace que el proceso de escalado vertical se aplique de inmediato. Para aplazar la operación de ampliación al siguiente periodo de mantenimiento, utilice `--no-apply-immediately`.

   Para Linux, macOS o Unix:

   ```
   aws elasticache modify-replication-group \
        --replication-group-id my-repl-group \
        --cache-node-type cache.t2.small  \
        --cache-parameter-group-name redis32-m3-2xl \
        --apply-immediately
   ```

   Para Windows:

   ```
   aws elasticache modify-replication-group ^
        --replication-group-id my-repl-group ^
        --cache-node-type cache.t2.small  ^
        --cache-parameter-group-name redis32-m3-2xl \
        --apply-immediately
   ```

   La salida de este comando tiene un aspecto similar al siguiente (formato JSON).

```
{"ReplicationGroup": {
         "Status": "available",
         "Description": "Some description",
         "NodeGroups": [
             {
                 "Status": "available",
                 "NodeGroupMembers": [
                     {
                         "CurrentRole": "primary",
                         "PreferredAvailabilityZone": "us-west-2b",
                         "CacheNodeId": "0001",
                         "ReadEndpoint": {
                             "Port": 6379,
                             "Address": "my-repl-
group-001.8fdx4s.0001.usw2.cache.amazonaws.com"
                         },
                         "CacheClusterId": "my-repl-group-001"
                     },
                     {
                         "CurrentRole": "replica",
                         "PreferredAvailabilityZone": "us-west-2c",
                         "CacheNodeId": "0001",
                         "ReadEndpoint": {
                             "Port": 6379,
                             "Address": "my-repl-
group-002.8fdx4s.0001.usw2.cache.amazonaws.com"
                         },
                         "CacheClusterId": "my-repl-group-002"
                     }
                 ],
                 "NodeGroupId": "0001",
                 "PrimaryEndpoint": {
                     "Port": 6379,
                     "Address": "my-repl-
group.8fdx4s.ng.0001.usw2.cache.amazonaws.com"
                 }
             }
         ],
         "ReplicationGroupId": "my-repl-group",
         "SnapshotRetentionLimit": 1,
         "AutomaticFailover": "disabled",
         "SnapshotWindow": "12:00-13:00",
```

```
            "SnapshottingClusterId": "my-repl-group-002",
            "MemberClusters": [
                "my-repl-group-001",
                "my-repl-group-002",
            ],
            "PendingModifiedValues": {}
        }
    }
```

Para obtener más información consulte modify-replication-group en la Referencia de la AWS CLI.

3. Si utilizó el parámetro `--apply-immediately`, monitorice el estado del grupo de reproducción utilizando el comando AWS CLI de la `describe-replication-group` con el siguiente parámetro. Cuando el estado cambie de modifying a available, podrá empezar a escribir en su nuevo grupo de replicación reducido.

   - `--replication-group-id`: nombre del grupo de replicación. Use este parámetro para describir un determinado grupo de replicación en lugar de todos los grupos de replicación.

   Para Linux, macOS o Unix:

   ```
   aws elasticache describe-replication-group \
        --replication-group-id my-replication-group
   ```

   Para Windows:

   ```
   aws elasticache describe-replication-groups ^
        --replication-group-id my-replication-group
   ```

   Para obtener más información, consulte describe-replication-groups en la Referencia de la AWS CLI.

Reducción vertical de un grupo de replicación de Valkey o Redis OSS (API de ElastiCache)

El siguiente proceso escala su grupo de reproducción de su tipo de nodo actual a un nuevo tipo de nodo más pequeño mediante la API de ElastiCache. Durante este proceso, ElastiCache actualiza las entradas de DNS para que apunten a los nuevos nodos. Por ello, no tiene que actualizar los puntos de enlace de su aplicación. En Valkey 7.2 y versiones posteriores o en Redis OSS 5.0.5 y

versiones posteriores, puede escalar clústeres con la conmutación por error habilitada mientras el clúster permanece en línea y atiende solicitudes de entrada. En Redis OSS 4.0.10 y versiones anteriores, puede observar una breve interrupción de las lecturas y escrituras del nodo principal mientras se actualiza la entrada de DNS. Sin embargo, las lecturas de los clústeres de réplica de lectura continuarán de forma ininterrumpida.

El tiempo que se tarda en la realización del escalado descendente a un tipo de nodo más pequeño varía en función de su tipo de nodo y de la cantidad de datos de su clúster actual.

Reducción vertical de un grupo de replicación de Valkey o Redis OSS (API de ElastiCache)

1. Determine a qué tipos de nodos puede reducir verticalmente mediante la acción `ListAllowedNodeTypeModifications` de la API de ElastiCache con el siguiente parámetro.

   - `ReplicationGroupId`: nombre del grupo de replicación. Use este parámetro para describir un grupo de replicación específico en lugar de todos los grupos de replicación.

   ```
   https://elasticache.us-west-2.amazonaws.com/
       ?Action=ListAllowedNodeTypeModifications
       &ReplicationGroupId=MyReplGroup
       &Version=2015-02-02
       &SignatureVersion=4
       &SignatureMethod=HmacSHA256
       &Timestamp=20150202T192317Z
       &X-Amz-Credential=<credential>
   ```

   Para obtener más información, consulte ListAllowedNodeTypeModifications en la Referencia de la API de Amazon ElastiCache.

2. Escale verticalmente su grupo de reproducción actual al nuevo tipo de nodo mediante la acción `ModifyReplicationGroup` de la API de ElastiCache y con los siguientes parámetros.

   - `ReplicationGroupId`: nombre del grupo de replicación.
   - `CacheNodeType`: nuevo tipo de nodo más pequeño de los clústeres en este grupo de replicación. Este valor debe ser uno de los tipos de instancia devueltos por la acción `ListAllowedNodeTypeModifications` en el paso anterior.
   - `CacheParameterGroupName`: [Opcional] Utilice este parámetro si va a utilizar `reserved-memory` para administrar la memoria reservada de su clúster. Especifique un grupo de parámetros de caché personalizado que reserve la cantidad correcta de memoria para

el nuevo tipo de nodo. Si va a utilizar `reserved-memory-percent`, puede omitir este parámetro.

- `ApplyImmediately`: establézcalo en `true` para que el proceso de escalado vertical se aplique de inmediato. Para aplazar el proceso de escalado descendente al siguiente periodo de mantenimiento, utilice `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=ModifyReplicationGroup
    &ApplyImmediately=true
    &CacheNodeType=cache.m3.2xlarge
    &CacheParameterGroupName=redis32-m3-2xl
    &ReplicationGroupId=myReplGroup
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &Version=2014-12-01
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20141201T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Para obtener más información, consulte ModifyReplicationGroup en la Referencia de la API de Amazon ElastiCache.

3. Si utilizó `ApplyImmediately=true`, monitoree el estado del grupo de reproducción mediante la acción `DescribeReplicationGroups` de la API de ElastiCache con los siguientes parámetros. Cuando el estado cambie de modifying a available, podrá empezar a escribir en su nuevo grupo de replicación reducido.

   - `ReplicationGroupId`: nombre del grupo de replicación. Use este parámetro para describir un determinado grupo de replicación en lugar de todos los grupos de replicación.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=DescribeReplicationGroups
    &ReplicationGroupId=MyReplGroup
    &Version=2015-02-02
    &SignatureVersion=4
```

```
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
    &X-Amz-Credential=<credential>
```

Para obtener más información, consulte DescribeReplicationGroups en la Referencia de la API de Amazon ElastiCache.

Aumento de la capacidad de lectura

Para aumentar la capacidad de lectura, añada réplicas de lectura (un máximo de cinco) a su grupo de replicación de Valkey o Redis OSS.

Puede escalar la capacidad de lectura de su clúster de Valkey o Redis OSS mediante la consola de ElastiCache, la AWS CLI o la API de ElastiCache. Para obtener más información, consulte Adición de una réplica de lectura para Valkey o Redis OSS (modo de clúster deshabilitado).

Reducción de la capacidad de lectura

Para reducir la capacidad de lectura, elimine una o varias réplicas de lectura de su clúster de Valkey o Redis OSS con réplicas (lo que se conoce como grupo de replicación en la API/CLI). Si el clúster tiene habilitado Multi-AZ con conmutación por error automática, no puede eliminar la última réplica de lectura sin deshabilitar primero Multi-AZ. Para obtener más información, consulte Modificación de un grupo de reproducción.

Para obtener más información, consulte Eliminación de una réplica de lectura para Valkey o Redis OSS (modo de clúster deshabilitado).

Escalado de clústeres de Valkey o Redis OSS (modo de clúster habilitado)

A medida que cambie la demanda en los clústeres, puede decidir mejorar el rendimiento o reducir los costos al cambiar el número de particiones en su clúster de Valkey o Redis OSS (modo de clúster habilitado). Recomendamos que utilice el escalado horizontal online, ya que permite que el clúster continúe sirviendo las solicitudes durante el proceso de escalado.

Entre las condiciones en las que puede decidir cambiar el escalado de su clúster se incluyen las siguientes:

- Presión de memoria:

  Si los nodos del clúster tienen presión de la memoria, puede decidir realizar un escalado ascendente para tener más recursos con el fin de almacenar los datos y servir las solicitudes mejor.

  Para determinar si los nodos se encuentran bajo presión de memoria, supervise las siguientes métricas: FreeableMemory, SwapUsage y BytesUsedForCache.
- Cuello de botella de CPU o de red:

  Si se producen muchos problemas de latencia o rendimiento en su clúster, tal vez deba hacer un escalado ascendente para resolverlos.

  Puede monitorear los niveles de latencia y rendimiento mediante el monitoreo de las siguientes métricas: CPUUtilization, NetworkBytesIn, NetworkBytesOut, CurrConnections y NewConnections.
- El escalado del clúster es excesivo:

  La demanda actual en su clúster es tal que el escalado descendente no afecta al rendimiento y reduce los costos.

  Puede supervisar el uso del clúster para determinar si puede realizar una reducción horizontal con las siguientes métricas: FreeableMemory, SwapUsage, BytesUsedForCache, CPUUtilization, NetworkBytesIn, NetworkBytesOut, CurrConnections y NewConnections.

Impacto de rendimiento del escalado

Cuando escala utilizando el proceso sin conexión, el clúster no está en línea durante una parte importante del proceso y, por tanto, no puede atender las solicitudes. Cuando escala utilizando el método online, como el escalado es una operación que realiza un uso intensivo de computación, se

deteriora algo el rendimiento, aunque el clúster sigue atendiendo las solicitudes en toda la operación de escalado. El nivel de deterioro de la experiencia depende del uso normal de la CPU y sus datos.

Existen dos formas de escalar el clúster de Valkey o Redis OSS (modo de clúster deshabilitado): el escalado horizontal y vertical.

- El escalado horizontal le permite cambiar el número de grupos de nodo (fragmentos) en el grupo de reproducción agregando o eliminando grupos de nodos (fragmentos). El proceso de partición en línea le permite escalar verticalmente/horizontalmente mientras el clúster sigue ofreciendo solicitudes entrantes.

  Configurar las ranuras en el nuevo clúster de forma diferente que en el clúster anterior. Solo método sin conexión.
- Escalado vertical: cambie el tipo de nodo para cambiar el tamaño del clúster. El proceso de escalado vertical online le permite el escalado ascendente y descendente mientras el clúster sigue ofreciendo solicitudes entrantes.

Si reduce el tamaño y la capacidad de memoria del clúster mediante la reducción horizontal o vertical, asegúrese de que la nueva configuración disponga de memoria suficiente para sus datos y para asumir la sobrecarga de Valkey o Redis OSS.

Para obtener más información, consulte Elección del tamaño del nodo.

Contenido

- Cambio de particiones sin conexión para Valkey o Redis OSS (modo de clúster habilitado)
- Cambio de particiones en línea para Valkey o Redis OSS (modo de clúster habilitado)
  - Adición de particiones con los cambios de particiones en línea
  - Eliminación de particiones con los cambios de particiones en línea
    - Eliminación de particiones (consola)
    - Eliminación de particiones (AWS CLI)
    - Eliminación de particiones (API de ElastiCache)
  - Reequilibrado de particiones en línea
    - Reequilibrado de fragmentos online (consola)
    - Reequilibrado de particiones en línea (AWS CLI)
    - Reequilibrio de particiones en línea (API de ElastiCache)

- Escalado vertical en línea mediante la modificación del tipo de nodo
  - Escalado vertical en línea
    - Escalado vertical de clústeres de Valkey o Redis OSS (consola)
    - Escalado vertical de clústeres de Valkey o Redis OSS (AWS CLI)
    - Escalado vertical de clústeres de Valkey o Redis OSS (API de ElastiCache)
  - Reducción vertical en línea
    - Reducción vertical de clústeres de Valkey o Redis OSS (consola)
    - Reducción vertical de clústeres de Valkey o Redis OSS (AWS CLI)
    - Reducción vertical de clústeres de Valkey o Redis OSS (API de ElastiCache)

Cambio de particiones sin conexión para Valkey o Redis OSS (modo de clúster habilitado)

La ventaja principal que obtiene de la reconfiguración de fragmentos sin conexión es que puede hacer algo más que agregar o eliminar fragmentos de su grupo de reproducción. Al hacer cambios de las particiones y reequilibrios sin conexión, además de cambiar el número de particiones en el grupo de replicación, puede hacer lo siguiente:

> **Note**
>
> Los cambios de particiones sin conexión no se admiten en los clústeres de Valkey o Redis OSS con la organización de datos en niveles habilitada. Para obtener más información, consulte Organización de datos en niveles en ElastiCache.

- Cambiar el tipo de nodo de su grupo de reproducción.
- Especificar la zona de disponibilidad de cada nodo del grupo de reproducción.
- Actualizar a una nueva versión del motor.
- Especificar el número de nodos de réplica de cada fragmento independientemente.
- Especificar el espacio de claves de cada fragmento.

La desventaja principal de la reconfiguración de particiones sin conexión es que el clúster está sin conexión al comentar la parte de restauración del proceso y así continuará hasta que actualice los puntos de conexión de la aplicación. El tiempo que el clúster está sin conexión varía según la cantidad de datos del clúster.

Reconfiguración de las particiones del clúster de Valkey o Redis OSS (modo de clúster habilitado) sin conexión

1. Cree una copia de seguridad manual del clúster de Valkey o Redis OSS existente. Para obtener más información, consulte Copias de seguridad manuales.
2. Cree un nuevo clúster restaurándolo a partir de esta copia de seguridad. Para obtener más información, consulte Restauración desde una copia de seguridad a una nueva caché.
3. Actualice los puntos de conexión de la aplicación a los puntos de conexión del nuevo clúster. Para obtener más información, consulte Búsqueda de puntos de conexión en ElastiCache.

Cambio de particiones en línea para Valkey o Redis OSS (modo de clúster habilitado)

Si realiza un nuevo cambio y reequilibrio de particiones en línea de las particiones con ElastiCache Valkey 7.2 o versiones posteriores, o Redis OSS versión 3.2.10 o posteriores, puede escalar Valkey o Redis OSS (modo de clúster habilitado) de forma dinámica sin tiempo de inactividad. Este enfoque significa que el clúster puede seguir atendiendo las solicitudes, incluso mientras esté en curso el escalado o el reequilibrado.

Puede hacer lo siguiente:

- Escalado horizontal: aumente la capacidad de lectura y escritura al añadir particiones (grupos de nodos) a su clúster (grupo de replicación) de Valkey o Redis OSS (modo de clúster habilitado).

  Si agrega uno o varios fragmentos a su grupo de reproducción, el número de nodos de cada nuevo fragmento es el mismo que el número de nodos en el menor de los fragmentos existentes.
- Reducción horizontal: reduzca la capacidad de lectura y escritura, y, por lo tanto, los costos, al eliminar particiones del clúster de Valkey o Redis OSS (modo de clúster habilitado).
- Reequilibrio: mueva los espacios de claves entre las particiones de su clúster de Valkey o Redis OSS (modo de clúster habilitado) de modo que se distribuyan de la manera más uniforme posible entre las particiones.

No puede hacer lo siguiente:

- Configurar los fragmentos de forma independiente:

  No puede especificar el espacio de claves para fragmentos de manera independiente. Para ello, debe utilizar el proceso sin conexión.

En la actualidad, las siguientes limitaciones se aplican a los cambios y el reequilibrado de particiones en línea de ElastiCache:

- Estos procesos requieren Valkey 7.2 y versiones posteriores o Redis OSS 3.2.10 o posterior. Para obtener información acerca de la actualización de la versión del motor, consulte Administración de versiones para ElastiCache.
- Existen limitaciones con ranuras o espacios de claves y grandes elementos:

  Si alguna de las claves de una partición contiene un elemento grande, la clave no se puede migrar a una partición nueva en el escalado ascendente o el reequilibrado. Esta funcionalidad puede provocar particiones desequilibradas.

  Si alguna de las claves de una partición contiene un elemento grande (elementos mayores que 256 MB después de la serialización), esa partición no se elimina en la reducción horizontal. Esta funcionalidad puede provocar que algunas particiones no se eliminen.
- Al realizar el escalado ascendente, el número de nodos de cualquier fragmento nuevo es igual al número de nodo del fragmento existente más pequeño.
- Al realizar el escalado ascendente, las etiquetas que son comunes a todos los fragmentos existentes se copian en los fragmentos nuevos.
- Al escalar horizontalmente un clúster de almacén de datos global, ElastiCache no replicará automáticamente las funciones de uno de los nodos existentes a los nodos nuevos. Recomendamos cargar las funciones en las particiones nuevas después de escalar horizontalmente el clúster para que todas las particiones tengan las mismas funciones.

> **Note**
>
> En ElastiCache para Valkey 7.2 y versiones posteriores y ElastiCache para Redis OSS versión 7 o posteriores: al escalar horizontalmente el clúster, ElastiCache replicará automáticamente las funciones cargadas en uno de los nodos existentes (seleccionado de forma aleatoria) en los nodos nuevos. Si la aplicación usa funciones, recomendamos cargar todas las funciones en todas las particiones antes de escalar horizontalmente para que el clúster no termine con diferentes definiciones de funciones en distintas particiones.

Para obtener más información, consulte Cambio de tamaño de clústeres online.

Puede escalar horizontalmente o reequilibrar sus clústeres de Valkey o Redis OSS (modo de clúster habilitado) mediante la Consola de administración de AWS, la AWS CLI y la API de ElastiCache.

Adición de particiones con los cambios de particiones en línea

Puede añadir particiones a su clúster de Valkey o Redis OSS (modo de clúster habilitado) mediante la Consola de administración de AWS, la AWS CLI o la API de ElastiCache. Al añadir particiones a un clúster de Valkey o Redis OSS (modo de clúster habilitado), las etiquetas en las particiones existentes se copian sobre las particiones nuevas.

Adición de particiones (consola)

Puede utilizar la Consola de administración de AWS para añadir una o varias particiones a su clúster de Valkey o Redis OSS (modo de clúster habilitado). El siguiente procedimiento describe el proceso.

Adición de particiones a su clúster de Valkey o Redis OSS (modo de clúster habilitado)

1. Abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Clústeres de Valkey o Clústeres de Redis OSS.
3. Localice y elija el nombre del clúster de Valkey o Redis OSS (modo de clúster habilitado) al que desea añadir particiones, pero no la casilla situada a la izquierda del nombre del clúster.

   > **Tip**
   >
   > En Valkey o Redis OSS (modo de clúster habilitado), aparecerá Valkey en clúster o Redis OSS en clúster en la columna Modo

4. Elija Add shards (Agregar fragmentos).
   a. Para Number of shards to be added, elija el número de fragmentos que desee agregar a este clúster.
   b. En Availability zone(s), elija No preference o Specify availability zones.
   c. Si ha elegido Specify availability zones, por cada nodo de cada fragmento, seleccione la zona de disponibilidad del nodo en la lista de zonas de disponibilidad.
   d. Elija Add (Agregar).

Adición de particiones (AWS CLI)

En el siguiente proceso, se describe cómo reconfigurar las particiones en su clúster de Valkey o Redis OSS (modo de clúster habilitado) al añadir particiones mediante la AWS CLI.

Use los siguientes parámetros con `modify-replication-group-shard-configuration`.

Parámetros

- `--apply-immediately`: obligatorio. Especifica que la operación de reconfiguración de fragmentos comience inmediatamente.
- `--replication-group-id`: obligatorio. Especifica en qué grupo de replicación (clúster) se debe realizar la operación de reconfiguración de fragmentos.
- `--node-group-count`: obligatorio. Especifica el número de fragmentos (grupos de nodos) que deben existir cuando se complete la operación. Al añadir fragmentos, el valor de `--node-group-count` debe ser mayor que el número actual de fragmentos.

  Si lo desea, puede especificar la zona de disponibilidad de cada nodo en el grupo de replicación con `--resharding-configuration`.
- `--resharding-configuration`: opcional. Una lista de zonas de disponibilidad preferidas para cada nodo en cada fragmento en el grupo de replicación. Utilice este parámetro solo si el valor de `--node-group-count` es mayor que el número actual de fragmentos. Si este parámetro se omite al agregar particiones, Amazon ElastiCache selecciona las zonas de disponibilidad de los nodos nuevos.

En el siguiente ejemplo, se reconfiguran los espacios de claves en cuatro particiones en el clúster de Valkey o Redis OSS (modo de clúster habilitado) denominado `my-cluster`. El ejemplo también especifica la zona de disponibilidad de cada nodo en cada fragmento. La operación comienza inmediatamente.

Example - Adición de fragmentos

Para Linux, macOS o Unix:

```
aws elasticache modify-replication-group-shard-configuration \
    --replication-group-id my-cluster \
    --node-group-count 4 \
    --resharding-configuration \
        "PreferredAvailabilityZones=us-east-2a,us-east-2c" \
```

```
        "PreferredAvailabilityZones=us-east-2b,us-east-2a" \
        "PreferredAvailabilityZones=us-east-2c,us-east-2d" \
        "PreferredAvailabilityZones=us-east-2d,us-east-2c" \
    --apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group-shard-configuration ^
    --replication-group-id my-cluster ^
    --node-group-count 4 ^
    --resharding-configuration ^
        "PreferredAvailabilityZones=us-east-2a,us-east-2c" ^
        "PreferredAvailabilityZones=us-east-2b,us-east-2a" ^
        "PreferredAvailabilityZones=us-east-2c,us-east-2d" ^
        "PreferredAvailabilityZones=us-east-2d,us-east-2c" ^
    --apply-immediately
```

Para obtener más información, consulte modify-replication-group-shard-configuration en la documentación de la AWS CLI.

Adición de particiones (API de ElastiCache)

Puede utilizar la API de ElastiCache para reconfigurar las particiones en su clúster de Valkey o Redis OSS (modo de clúster habilitado) en línea mediante la operación `ModifyReplicationGroupShardConfiguration`.

Use los siguientes parámetros con `ModifyReplicationGroupShardConfiguration`.

Parámetros

- `ApplyImmediately=true`: obligatorio. Especifica que la operación de reconfiguración de fragmentos comience inmediatamente.
- `ReplicationGroupId`: obligatorio. Especifica en qué grupo de replicación (clúster) se debe realizar la operación de reconfiguración de fragmentos.
- `NodeGroupCount`: obligatorio. Especifica el número de fragmentos (grupos de nodos) que deben existir cuando se complete la operación. Al añadir fragmentos, el valor de `NodeGroupCount` debe ser mayor que el número actual de fragmentos.

  Si lo desea, puede especificar la zona de disponibilidad de cada nodo en el grupo de replicación con `ReshardingConfiguration`.

- `ReshardingConfiguration`: opcional. Una lista de zonas de disponibilidad preferidas para cada nodo en cada fragmento en el grupo de replicación. Utilice este parámetro solo si el valor de `NodeGroupCount` es mayor que el número actual de fragmentos. Si este parámetro se omite al agregar particiones, Amazon ElastiCache selecciona las zonas de disponibilidad de los nodos nuevos.

En el siguiente proceso, se describe cómo reconfigurar las particiones en su clúster de Valkey o Redis OSS (modo de clúster habilitado) al añadir particiones mediante la API de ElastiCache.

Example - Adición de fragmentos

En el siguiente ejemplo, se añaden grupos de nodos al clúster de Valkey o Redis OSS (modo de clúster habilitado) `my-cluster`, por lo que hay un total de cuatro grupos de nodos cuando se completa la operación. El ejemplo también especifica la zona de disponibilidad de cada nodo en cada fragmento. La operación comienza inmediatamente.

```
https://elasticache.us-east-2.amazonaws.com/
    ?Action=ModifyReplicationGroupShardConfiguration
    &ApplyImmediately=true
    &NodeGroupCount=4
    &ReplicationGroupId=my-cluster

 &ReshardingConfiguration.ReshardingConfiguration.1.PreferredAvailabilityZones.AvailabilityZone
east-2a

 &ReshardingConfiguration.ReshardingConfiguration.1.PreferredAvailabilityZones.AvailabilityZone
east-2c

 &ReshardingConfiguration.ReshardingConfiguration.2.PreferredAvailabilityZones.AvailabilityZone
east-2b

 &ReshardingConfiguration.ReshardingConfiguration.2.PreferredAvailabilityZones.AvailabilityZone
east-2a

 &ReshardingConfiguration.ReshardingConfiguration.3.PreferredAvailabilityZones.AvailabilityZone
east-2c

 &ReshardingConfiguration.ReshardingConfiguration.3.PreferredAvailabilityZones.AvailabilityZone
east-2d
```

```
 &ReshardingConfiguration.ReshardingConfiguration.4.PreferredAavailabilityZones.AvailabilityZone
east-2d

 &ReshardingConfiguration.ReshardingConfiguration.4.PreferredAvailabilityZones.AvailabilityZone
east-2c
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20171002T192317Z
    &X-Amz-Credential=<credential>
```

Para obtener más información, consulte ModifyReplicationGroupShardConfiguration en la Referencia de la API de ElastiCache.

Eliminación de particiones con los cambios de particiones en línea

Puede eliminar particiones de su clúster de Valkey o Redis OSS (modo de clúster habilitado) mediante la Consola de administración de AWS, la AWS CLI o la API de ElastiCache.

Temas

- Eliminación de particiones (consola)
- Eliminación de particiones (AWS CLI)
- Eliminación de particiones (API de ElastiCache)

Eliminación de particiones (consola)

En el siguiente proceso, se describe cómo reconfigurar las particiones en su clúster de Valkey o Redis OSS (modo de clúster habilitado) al eliminar particiones mediante la Consola de administración de AWS.

Antes de eliminar grupos de nodos (particiones) de su grupo de reproducción, ElastiCache comprueba que todos los datos van a caber en las particiones restantes. Si los datos caben, los fragmentos especificados se eliminan del grupo de reproducción según lo solicitado. Si los datos no van a caber en los grupos de nodos restantes, el proceso se termina y el grupo de reproducción se deja con la misma configuración de nodo de antes de que se hiciera la solicitud.

Puede utilizar la Consola de administración de AWS para eliminar una o varias particiones de su clúster de Valkey o Redis OSS (modo de clúster habilitado). No puede eliminar todos los fragmentos

de un grupo de replicación. En su lugar, debe eliminar el grupo de reproducción. Para obtener más información, consulte Eliminación de un grupo de reproducción. El siguiente procedimiento describe el proceso para eliminar uno o varios fragmentos.

Eliminación de particiones de su clúster de Valkey o Redis OSS (modo de clúster habilitado)

1. Abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Clústeres de Valkey o Clústeres de Redis OSS.
3. Localice y elija el nombre del clúster de Valkey o Redis OSS (modo de clúster habilitado) del que desea eliminar particiones, pero no la casilla situada a la izquierda del nombre del clúster.

   > **Tip**
   >
   > Los clústeres de Valkey o Redis OSS (modo de clúster habilitado) tienen un valor de 1 o superior en la columna Particiones.

4. En la lista de fragmentos, elija la casilla situada a la izquierda del nombre de cada fragmento que desee eliminar.
5. Elija Delete shard (Eliminar fragmentos).

Eliminación de particiones (AWS CLI)

En el siguiente proceso, se describe cómo reconfigurar las particiones en su clúster de Valkey o Redis OSS (modo de clúster habilitado) al eliminar particiones mediante la AWS CLI.

> **Important**
>
> Antes de eliminar grupos de nodos (particiones) de su grupo de reproducción, ElastiCache comprueba que todos los datos van a caber en las particiones restantes. Si los datos caben, los fragmentos especificados (`--node-groups-to-remove`) se eliminan del grupo de reproducción según lo solicitado y sus espacios de clave se asignan a los fragmentos restantes. Si los datos no van a caber en los grupos de nodos restantes, el proceso se termina y el grupo de reproducción se deja con la misma configuración de nodo de antes de que se hiciera la solicitud.

Puede utilizar la AWS CLI para eliminar una o varias particiones de su clúster de Valkey o Redis OSS (modo de clúster habilitado). No puede eliminar todos los fragmentos de un grupo de

replicación. En su lugar, debe eliminar el grupo de reproducción. Para obtener más información, consulte Eliminación de un grupo de reproducción.

Use los siguientes parámetros con `modify-replication-group-shard-configuration`.

Parámetros

- `--apply-immediately`: obligatorio. Especifica que la operación de reconfiguración de fragmentos comience inmediatamente.
- `--replication-group-id`: obligatorio. Especifica en qué grupo de replicación (clúster) se debe realizar la operación de reconfiguración de fragmentos.
- `--node-group-count`: obligatorio. Especifica el número de fragmentos (grupos de nodos) que deben existir cuando se complete la operación. Al eliminar fragmentos, el valor de `--node-group-count` debe ser menor que el número actual de fragmentos.

- `--node-groups-to-remove`: obligatorio cuando `--node-group-count` es menor que el número actual de grupos de nodos (particiones). Una lista de ID de fragmento (grupo de nodos) para eliminar del grupo de replicación.

El siguiente procedimiento describe el proceso para eliminar uno o varios fragmentos.

Example - Eliminación de fragmentos

En el siguiente ejemplo, se eliminan dos grupos de nodos del clúster `my-cluster` de Valkey o Redis OSS (modo de clúster habilitado), por lo que hay un total de dos grupos de nodos cuando se completa la operación. Los espacios de claves de los fragmentos eliminados se distribuyen de manera uniforme en los fragmentos restantes.

Para Linux, macOS o Unix:

```
aws elasticache modify-replication-group-shard-configuration \
    --replication-group-id my-cluster \
    --node-group-count 2 \
    --node-groups-to-remove "0002" "0003" \
    --apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group-shard-configuration ^
```

```
    --replication-group-id my-cluster ^
    --node-group-count 2 ^
    --node-groups-to-remove "0002" "0003" ^
    --apply-immediately
```

Eliminación de particiones (API de ElastiCache)

Puede utilizar la API de ElastiCache para reconfigurar las particiones en su clúster de Valkey o Redis OSS (modo de clúster habilitado) en línea mediante la operación `ModifyReplicationGroupShardConfiguration`.

En el siguiente proceso, se describe cómo reconfigurar las particiones en su clúster de Valkey o Redis OSS (modo de clúster habilitado) al eliminar particiones mediante la API de ElastiCache.

> **Important**
>
> Antes de eliminar grupos de nodos (particiones) de su grupo de reproducción, ElastiCache comprueba que todos los datos van a caber en las particiones restantes. Si los datos caben, los fragmentos especificados (`NodeGroupsToRemove`) se eliminan del grupo de reproducción según lo solicitado y sus espacios de clave se asignan a los fragmentos restantes. Si los datos no van a caber en los grupos de nodos restantes, el proceso se termina y el grupo de reproducción se deja con la misma configuración de nodo de antes de que se hiciera la solicitud.

Puede utilizar la API de ElastiCache para eliminar una o varias particiones de su clúster de Valkey o Redis OSS (modo de clúster habilitado). No puede eliminar todos los fragmentos de un grupo de replicación. En su lugar, debe eliminar el grupo de reproducción. Para obtener más información, consulte Eliminación de un grupo de reproducción.

Use los siguientes parámetros con `ModifyReplicationGroupShardConfiguration`.

Parámetros

- `ApplyImmediately=true`: obligatorio. Especifica que la operación de reconfiguración de fragmentos comience inmediatamente.
- `ReplicationGroupId`: obligatorio. Especifica en qué grupo de replicación (clúster) se debe realizar la operación de reconfiguración de fragmentos.

- `NodeGroupCount`: obligatorio. Especifica el número de fragmentos (grupos de nodos) que deben existir cuando se complete la operación. Al eliminar fragmentos, el valor de `NodeGroupCount` debe ser menor que el número actual de fragmentos.
- `NodeGroupsToRemove`: obligatorio cuando `--node-group-count` es menor que el número actual de grupos de nodos (particiones). Una lista de ID de fragmento (grupo de nodos) para eliminar del grupo de replicación.

El siguiente procedimiento describe el proceso para eliminar uno o varios fragmentos.

Example - Eliminación de fragmentos

En el siguiente ejemplo, se eliminan dos grupos de nodos del clúster `my-cluster` de Valkey o Redis OSS (modo de clúster habilitado), por lo que hay un total de dos grupos de nodos cuando se completa la operación. Los espacios de claves de los fragmentos eliminados se distribuyen de manera uniforme en los fragmentos restantes.

```
https://elasticache.us-east-2.amazonaws.com/
    ?Action=ModifyReplicationGroupShardConfiguration
    &ApplyImmediately=true
    &NodeGroupCount=2
    &ReplicationGroupId=my-cluster
    &NodeGroupsToRemove.member.1=0002
    &NodeGroupsToRemove.member.2=0003
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20171002T192317Z
    &X-Amz-Credential=<credential>
```

Reequilibrado de particiones en línea

Puede reequilibrar particiones de su clúster de Valkey o Redis OSS (modo de clúster habilitado) mediante la Consola de administración de AWS, la AWS CLI o la API de ElastiCache.

Temas

- Reequilibrado de fragmentos online (consola)
- Reequilibrado de particiones en línea (AWS CLI)
- Reequilibrio de particiones en línea (API de ElastiCache)

Reequilibrado de fragmentos online (consola)

En el siguiente proceso, se describe cómo reconfigurar las particiones en su clúster de Valkey o Redis OSS (modo de clúster habilitado) al reequilibrar particiones mediante la Consola de administración de AWS.

Reequilibrado de espacios de claves entre las particiones de su clúster de Valkey o Redis OSS (modo de clúster habilitado)

1. Abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Clústeres de Valkey o Clústeres de Redis OSS.
3. Elija el nombre del clúster de Valkey o Redis OSS (modo de clúster habilitado) que desee reequilibrar, pero no la casilla situada a la izquierda del nombre del clúster.

   > **Tip**
   >
   > Los clústeres de Valkey o Redis OSS (modo de clúster habilitado) tienen un valor de 1 o superior en la columna Particiones.

4. Elija Rebalance.
5. Cuando se solicite, elija Rebalance. Puede ver un mensaje similar a este: *Slots in the replication group are uniformly distributed (Las ranuras del grupo de reproducción se distribuyen de forma uniforme). Nothing to do. (Service: AmazonElastiCache; Status Code: 400; Error Code: InvalidReplicationGroupState; Request ID: 2246cebd-9721-11e7-8d5b-e1b0f086c8cf)*. Si lo ve, elija Cancel.

Reequilibrado de particiones en línea (AWS CLI)

Use los siguientes parámetros con `modify-replication-group-shard-configuration`.

Parámetros

- `-apply-immediately`: obligatorio. Especifica que la operación de reconfiguración de fragmentos comience inmediatamente.
- `--replication-group-id`: obligatorio. Especifica en qué grupo de replicación (clúster) se debe realizar la operación de reconfiguración de fragmentos.

- `--node-group-count`: obligatorio. Para reequilibrar los espacios de claves en todos los fragmentos del clúster, este valor debe ser igual que el número de fragmentos.

En el siguiente proceso, se describe cómo reconfigurar las particiones en su clúster de Valkey o Redis OSS (modo de clúster habilitado) al reequilibrar particiones mediante la AWS CLI.

Example - Reequilibrado de los fragmentos en un clúster

En el siguiente ejemplo, se reequilibran las ranuras en el clúster `my-cluster` de Valkey o Redis OSS (modo de clúster habilitado) para que las ranuras se distribuyan del modo más uniforme posible. El valor de `--node-group-count` (4) es el número de fragmentos que hay actualmente en el clúster.

Para Linux, macOS o Unix:

```
aws elasticache modify-replication-group-shard-configuration \
    --replication-group-id my-cluster \
    --node-group-count 4 \
    --apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group-shard-configuration ^
    --replication-group-id my-cluster ^
    --node-group-count 4 ^
    --apply-immediately
```

Reequilibrio de particiones en línea (API de ElastiCache)

Puede utilizar la API de ElastiCache para reconfigurar las particiones en su clúster de Valkey o Redis OSS (modo de clúster habilitado) en línea mediante la operación `ModifyReplicationGroupShardConfiguration`.

Use los siguientes parámetros con `ModifyReplicationGroupShardConfiguration`.

Parámetros

- `ApplyImmediately=true`: obligatorio. Especifica que la operación de reconfiguración de fragmentos comience inmediatamente.
- `ReplicationGroupId`: obligatorio. Especifica en qué grupo de replicación (clúster) se debe realizar la operación de reconfiguración de fragmentos.

- `NodeGroupCount`: obligatorio. Para reequilibrar los espacios de claves en todos los fragmentos del clúster, este valor debe ser igual que el número de fragmentos.

En el siguiente proceso, se describe cómo reconfigurar las particiones en su clúster de Valkey o Redis OSS (modo de clúster habilitado) al reequilibrar particiones mediante la API de ElastiCache.

Example - Reequilibrado de un clúster

En el siguiente ejemplo, se reequilibran las ranuras en el clúster `my-cluster` de Valkey o Redis OSS (modo de clúster habilitado) para que las ranuras se distribuyan del modo más uniforme posible. El valor de `NodeGroupCount` (4) es el número de fragmentos que hay actualmente en el clúster.

```
https://elasticache.us-east-2.amazonaws.com/
    ?Action=ModifyReplicationGroupShardConfiguration
    &ApplyImmediately=true
    &NodeGroupCount=4
    &ReplicationGroupId=my-cluster
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20171002T192317Z
    &X-Amz-Credential=<credential>
```

Escalado vertical en línea mediante la modificación del tipo de nodo

Al utilizar el escalado vertical en línea con Valkey versión 7.2 o posteriores, o Redis OSS versión 3.2.10 o posteriores, puede escalar sus clústeres de Valkey o Redis OSS dinámicamente con el tiempo de inactividad mínimo. Esto permite que el clúster de Valkey o Redis OSS atienda solicitudes incluso mientras se escala.

> **Note**
>
> No se admite el escalado entre un clúster de organización de datos en niveles (por ejemplo, un clúster que utiliza un tipo de nodo r6gd) y un clúster que no utiliza la organización de datos en niveles (por ejemplo, un clúster que utiliza un tipo de nodo r6g). Para obtener más información, consulte Organización de datos en niveles en ElastiCache.

Puede hacer lo siguiente:

- Escalado vertical: aumente la capacidad de lectura y escritura ajustando el tipo de nodo del clúster de Valkey o Redis OSS para utilizar un tipo de nodo más grande.

  ElastiCache redimensiona de forma dinámica su clúster mientras permanece en línea y atiende solicitudes.
- Reducción vertical: reduzca verticalmente la capacidad de lectura y escritura al ajustar el tipo de nodo para utilizar un nodo más pequeño. Nuevamente, ElastiCache redimensiona de forma dinámica su clúster mientras permanece en línea y atiende solicitudes. En este caso, reduzca los costos reduciendo el tamaño del nodo.

> **Note**
>
> Los procesos de escalado ascendente y descendente dependen de la creación de clústeres con tipos de nodo seleccionados recientemente y la sincronización de los nuevos nodos con los anteriores. Para garantizar un flujo de escalado ascendente/descendente uniforme, realice el siguiente procedimiento:
>
> - Asegúrese de tener suficiente capacidad ENI (Elastic Network Interface, interfaz de red elástica). Si se aplica el escalado descendiente, asegúrese de que el nodo de tamaño más reducido tenga suficiente memoria para absorber el tráfico esperado.
>
>   Consulte para conocer las prácticas recomendadas sobre la administración de memoria Administración de la memoria reservada para Valkey y Redis OSS.
> - Aunque el proceso de escalado vertical está diseñado para que sea completamente online, se basa en la sincronización de datos entre el nodo antiguo y el nuevo. Recomendamos que inicie el escalado ascendente/descendente durante las horas en las que espera que el tráfico de datos sea mínimo.
> - Pruebe el comportamiento de la aplicación durante el escalado en un entorno de ensayo, si es posible.

Contenido

- Escalado vertical en línea
  - Escalado vertical de clústeres de Valkey o Redis OSS (consola)
  - Escalado vertical de clústeres de Valkey o Redis OSS (AWS CLI)
  - Escalado vertical de clústeres de Valkey o Redis OSS (API de ElastiCache)

- Reducción vertical en línea
  - Reducción vertical de clústeres de Valkey o Redis OSS (consola)
  - Reducción vertical de clústeres de Valkey o Redis OSS (AWS CLI)
  - Reducción vertical de clústeres de Valkey o Redis OSS (API de ElastiCache)

Escalado vertical en línea

Temas

- Escalado vertical de clústeres de Valkey o Redis OSS (consola)
- Escalado vertical de clústeres de Valkey o Redis OSS (AWS CLI)
- Escalado vertical de clústeres de Valkey o Redis OSS (API de ElastiCache)

Escalado vertical de clústeres de Valkey o Redis OSS (consola)

El siguiente procedimiento describe cómo escalar verticalmente un clúster de Valkey o Redis OSS mediante la consola de administración de ElastiCache. Durante este proceso, el clúster seguirá atendiendo solicitudes con un tiempo de inactividad mínimo.

Escalado vertical de un clúster de Valkey o Redis OSS (consola)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Clústeres de Valkey o Clústeres de Redis OSS.
3. En la lista de clústeres, elija el clúster.
4. Elija Modificar.
5. En el asistente Modify Cluster:
   - Elija el tipo de nodo que desee ampliar en la lista Node type. Para aplicar el escalado ascendente, seleccione un tipo de nodo superior a su nodo existente.
6. Si desea realizar el proceso de escalado ascendente de forma inmediata, elija el cuadro Apply immediately (Aplicar inmediatamente). Si no selecciona el cuadro Apply immediately, el proceso de ampliación se realizará durante el siguiente periodo de mantenimiento de este clúster.
7. Elija Modify (Modificar).

Si eligió Apply immediately (Aplicar inmediatamente) en el paso anterior, el estado del clúster cambiará a modifying (en modificación). Cuando el estado cambie a available (disponible), la modificación se habrá completado y podrá empezar a utilizar el nuevo clúster.

Escalado vertical de clústeres de Valkey o Redis OSS (AWS CLI)

El siguiente procedimiento describe cómo escalar verticalmente un clúster de Valkey o Redis OSS mediante la AWS CLI. Durante este proceso, el clúster seguirá atendiendo solicitudes con un tiempo de inactividad mínimo.

Cómo escalar verticalmente un clúster de Valkey o Redis OSS (AWS CLI)

1. Determine los tipos de nodos a los que puede ampliar ejecutando el comando AWS CLI de la `list-allowed-node-type-modifications` con el siguiente parámetro.

   Para Linux, macOS o Unix:

   ```
   aws elasticache list-allowed-node-type-modifications \
        --replication-group-id my-replication-group-id
   ```

   Para Windows:

   ```
   aws elasticache list-allowed-node-type-modifications ^
        --replication-group-id my-replication-group-id
   ```

   La salida del comando anterior es similar a la siguiente (formato JSON).

   ```
   {
        "ScaleUpModifications": [
            "cache.m3.2xlarge",
            "cache.m3.large",
            "cache.m3.xlarge",
            "cache.m4.10xlarge",
            "cache.m4.2xlarge",
            "cache.m4.4xlarge",
            "cache.m4.large",
            "cache.m4.xlarge",
            "cache.r3.2xlarge",
            "cache.r3.4xlarge",
   ```

```
          "cache.r3.8xlarge",
          "cache.r3.large",
          "cache.r3.xlarge"
     ]
        "ScaleDownModifications": [
          "cache.t2.micro",
          "cache.t2.small ",
          "cache.t2.medium",
          "cache.t1.small "
     ],
}
```

Para obtener más información consulte list-allowed-node-type-modifications en la Referencia de la AWS CLI.

2. Modifique su grupo de reproducción para aplicar el escalado ascendente al nuevo tipo de nodo más amplio con el comando AWS CLI de la `modify-replication-group` y los siguientes parámetros.

   - `--replication-group-id`: el nombre del grupo de reproducción al que va a escalar verticalmente.
   - `--cache-node-type`: tipo de nodo nuevo al que desea escalar el clúster. Este valor debe ser uno de los tipos de nodos devueltos por el comando `list-allowed-node-type-modifications` en el paso 1.
   - `--cache-parameter-group-name`: [Opcional] Utilice este parámetro si va a utilizar `reserved-memory` para administrar la memoria reservada de su clúster. Especifique un grupo de parámetros de caché personalizado que reserve la cantidad correcta de memoria para el nuevo tipo de nodo. Si va a utilizar `reserved-memory-percent`, puede omitir este parámetro.
   - `--apply-immediately`: hace que el proceso de escalado vertical se aplique de inmediato. Para aplazar el proceso de ampliación al siguiente periodo de mantenimiento del clúster, utilice el parámetro `--no-apply-immediately`.

   Para Linux, macOS o Unix:

```
aws elasticache modify-replication-group  \
     --replication-group-id my-redis-cluster \
     --cache-node-type cache.m3.xlarge \
```

```
    --apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group ^
    --replication-group-id my-redis-cluster ^
    --cache-node-type cache.m3.xlarge ^
    --apply-immediately
```

La salida del comando anterior es similar a la siguiente (formato JSON).


```
{
  "ReplicationGroup": {
        "Status": "modifying",
        "Description": "my-redis-cluster",
        "NodeGroups": [
            {
                "Status": "modifying",
                "Slots": "0-16383",
                "NodeGroupId": "0001",
                "NodeGroupMembers": [
                    {
                        "PreferredAvailabilityZone": "us-east-1f",
                        "CacheNodeId": "0001",
                        "CacheClusterId": "my-redis-cluster-0001-001"
                    },
                    {
                        "PreferredAvailabilityZone": "us-east-1d",
                        "CacheNodeId": "0001",
                        "CacheClusterId": "my-redis-cluster-0001-002"
                    }
                ]
            }
        ],
        "ConfigurationEndpoint": {
            "Port": 6379,
            "Address": "my-redis-
cluster.r7gdfi.clustercfg.use1.cache.amazonaws.com"
        },
        "ClusterEnabled": true,
        "ReplicationGroupId": "my-redis-cluster",
```

```
        "SnapshotRetentionLimit": 1,
        "AutomaticFailover": "enabled",
        "SnapshotWindow": "07:30-08:30",
        "MemberClusters": [
            "my-redis-cluster-0001-001",
            "my-redis-cluster-0001-002"
        ],
        "CacheNodeType": "cache.m3.xlarge",
         "DataTiering": "disabled"
        "PendingModifiedValues": {}
    }
}
```

Para obtener más información consulte modify-replication-group en la Referencia de la AWS CLI.

3. Si ha utilizado `--apply-immediately`, compruebe el estado del clúster utilizando el comando `describe-cache-clusters` de la AWS CLIcon el siguiente parámetro. Cuando el estado cambie a disponible, podrá comenzar a usar el nuevo nodo de clúster de la caché de mayor tamaño.

Escalado vertical de clústeres de Valkey o Redis OSS (API de ElastiCache)

El siguiente proceso escala su clúster de su tipo de nodo actual a un nuevo tipo de nodo más grande mediante la API de ElastiCache. Durante este proceso, ElastiCache actualiza las entradas de DNS para que apunten a los nuevos nodos. Por ello, no tiene que actualizar los puntos de enlace de su aplicación. En Valkey 7.2 y versiones posteriores y en Redis OSS 5.0.5 y versiones posteriores, puede escalar clústeres con la conmutación por error habilitada mientras el clúster permanece en línea y atiende solicitudes de entrada. En Redis OSS 4.0.10 y versiones anteriores, puede observar una breve interrupción de las lecturas y escrituras del nodo principal mientras se actualiza la entrada de DNS.

El tiempo que se tarda en el escalado vertical a un tipo de nodo más grande varía en función de su tipo de nodo y de la cantidad de datos de su clúster actual.

Escalado vertical de un clúster de caché de Valkey o Redis OSS (API de ElastiCache)

1. Determine a qué tipos de nodos puede escalar verticalmente mediante la acción `ListAllowedNodeTypeModifications` de la API de ElastiCache con el siguiente parámetro.

- `ReplicationGroupId`: nombre del grupo de replicación. Use este parámetro para describir un grupo de replicación específico en lugar de todos los grupos de replicación.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=ListAllowedNodeTypeModifications
    &ReplicationGroupId=MyReplGroup
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
    &X-Amz-Credential=<credential>
```

Para obtener más información, consulte ListAllowedNodeTypeModifications en la Referencia de la API de Amazon ElastiCache.

2. Escale verticalmente su grupo de reproducción actual al nuevo tipo de nodo mediante la acción `ModifyReplicationGroup` de la API de ElastiCache y con los siguientes parámetros.

   - `ReplicationGroupId`: nombre del grupo de replicación.
   - `CacheNodeType`: nuevo tipo de nodo más grande de los clústeres en este grupo de replicación. Este valor debe ser uno de los tipos de instancia devueltos por la acción `ListAllowedNodeTypeModifications` en el paso anterior.
   - `CacheParameterGroupName`: [Opcional] Utilice este parámetro si va a utilizar `reserved-memory` para administrar la memoria reservada de su clúster. Especifique un grupo de parámetros de caché personalizado que reserve la cantidad correcta de memoria para el nuevo tipo de nodo. Si va a utilizar `reserved-memory-percent`, puede omitir este parámetro.
   - `ApplyImmediately`: establézcalo en `true` para que el proceso de escalado vertical se aplique de inmediato. Para aplazar el proceso de ampliación al siguiente periodo de mantenimiento, utilice `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=ModifyReplicationGroup
    &ApplyImmediately=true
    &CacheNodeType=cache.m3.2xlarge
    &CacheParameterGroupName=redis32-m3-2xl
    &ReplicationGroupId=myReplGroup
```

```
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &Version=2014-12-01
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20141201T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Para obtener más información, consulte ModifyReplicationGroup en la Referencia de la API de Amazon ElastiCache.

3. Si utilizó `ApplyImmediately=true`, monitoree el estado del grupo de reproducción mediante la acción `DescribeReplicationGroups` de la API de ElastiCache con los siguientes parámetros. Cuando el estado cambie de modifying a available, podrá empezar a escribir en su nuevo grupo de replicación ampliado.

   - `ReplicationGroupId`: nombre del grupo de replicación. Use este parámetro para describir un determinado grupo de replicación en lugar de todos los grupos de replicación.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=DescribeReplicationGroups
    &ReplicationGroupId=MyReplGroup
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
    &X-Amz-Credential=<credential>
```

Para obtener más información, consulte DescribeReplicationGroups en la Referencia de la API de Amazon ElastiCache.

Reducción vertical en línea

Temas

- Reducción vertical de clústeres de Valkey o Redis OSS (consola)
- Reducción vertical de clústeres de Valkey o Redis OSS (AWS CLI)

- Reducción vertical de clústeres de Valkey o Redis OSS (API de ElastiCache)

Reducción vertical de clústeres de Valkey o Redis OSS (consola)

El siguiente procedimiento describe cómo reducir verticalmente un clúster de Valkey o Redis OSS mediante la consola de administración de ElastiCache. Durante este proceso, el clúster de Valkey o Redis OSS seguirá atendiendo solicitudes con un tiempo de inactividad mínimo.

Reducción vertical de un clúster de Valkey o Redis OSS (consola)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Clústeres de Valkey o Clústeres de Redis OSS.
3. En la lista de clústeres, elija el clúster preferido.
4. Elija Modificar.
5. En el asistente Modify Cluster:
   - Elija el tipo de nodo que desee ampliar en la lista Node type. Para aplicar el escalado descendente, seleccione un tipo de nodo inferior a su nodo existente. Tenga en cuenta que no todos los tipos de nodos están disponibles para el proceso de reducción.
6. Si desea realizar el proceso de escalado descendente de forma inmediata, elija el cuadro Apply immediately (Aplicar inmediatamente). Si no selecciona el cuadro Apply immediately (Aplicar inmediatamente), el proceso de escalado descendente se realizará durante el siguiente periodo de mantenimiento de este clúster.
7. Elija Modify (Modificar).

   Si eligió Apply immediately (Aplicar inmediatamente) en el paso anterior, el estado del clúster cambiará a modifying (en modificación). Cuando el estado cambie a available (disponible), la modificación se habrá completado y podrá empezar a utilizar el nuevo clúster.

Reducción vertical de clústeres de Valkey o Redis OSS (AWS CLI)

El siguiente procedimiento describe cómo reducir verticalmente un clúster de Valkey o Redis OSS mediante la AWS CLI. Durante este proceso, el clúster seguirá atendiendo solicitudes con un tiempo de inactividad mínimo.

Cómo reducir verticalmente un clúster de Valkey o Redis OSS (AWS CLI)

1. Determine los tipos de nodos a los que puede aplicar el escalado descendente ejecutando el comando AWS CLI de la `list-allowed-node-type-modifications` con el siguiente parámetro.

   Para Linux, macOS o Unix:

   ```
   aws elasticache list-allowed-node-type-modifications \
        --replication-group-id my-replication-group-id
   ```

   Para Windows:

   ```
   aws elasticache list-allowed-node-type-modifications ^
        --replication-group-id my-replication-group-id
   ```

   La salida del comando anterior es similar a la siguiente (formato JSON).

   ```
   {
        "ScaleUpModifications": [
            "cache.m3.2xlarge",
            "cache.m3.large",
            "cache.m3.xlarge",
            "cache.m4.10xlarge",
            "cache.m4.2xlarge",
            "cache.m4.4xlarge",
            "cache.m4.large",
            "cache.m4.xlarge",
            "cache.r3.2xlarge",
            "cache.r3.4xlarge",
            "cache.r3.8xlarge",
            "cache.r3.large",
            "cache.r3.xlarge"
        ]

           "ScaleDownModifications": [
            "cache.t2.micro",
            "cache.t2.small ",
            "cache.t2.medium ",
            "cache.t1.small"
        ]
   ```

```
}
```

Para obtener más información consulte list-allowed-node-type-modifications en la Referencia de la AWS CLI.

2. Modifique su grupo de reproducción para aplicar el escalado descendente al nuevo tipo de nodo más pequeño con el comando AWS CLI de la `modify-replication-group` y los siguientes parámetros.

   - `--replication-group-id`: el nombre del grupo de reproducción al que va a reducir verticalmente.
   - `--cache-node-type`: tipo de nodo nuevo al que desea escalar el clúster. Este valor debe ser uno de los tipos de nodos devueltos por el comando `list-allowed-node-type-modifications` en el paso 1.
   - `--cache-parameter-group-name`: [Opcional] Utilice este parámetro si va a utilizar `reserved-memory` para administrar la memoria reservada de su clúster. Especifique un grupo de parámetros de caché personalizado que reserve la cantidad correcta de memoria para el nuevo tipo de nodo. Si va a utilizar `reserved-memory-percent`, puede omitir este parámetro.
   - `--apply-immediately`: hace que el proceso de escalado vertical se aplique de inmediato. Para aplazar el proceso de reducción al siguiente periodo de mantenimiento del clúster, utilice el parámetro `--no-apply-immediately`.

   Para Linux, macOS o Unix:

   ```
   aws elasticache modify-replication-group  \
        --replication-group-id my-redis-cluster \
        --cache-node-type cache.t2.micro \
        --apply-immediately
   ```

   Para Windows:

   ```
   aws elasticache modify-replication-group ^
        --replication-group-id my-redis-cluster ^
        --cache-node-type cache.t2.micro ^
        --apply-immediately
   ```

   La salida del comando anterior es similar a la siguiente (formato JSON).

```
{
  "ReplicationGroup": {
        "Status": "modifying",
        "Description": "my-redis-cluster",
        "NodeGroups": [
            {
                "Status": "modifying",
                "Slots": "0-16383",
                "NodeGroupId": "0001",
                "NodeGroupMembers": [
                    {
                        "PreferredAvailabilityZone": "us-east-1f",
                        "CacheNodeId": "0001",
                        "CacheClusterId": "my-redis-cluster-0001-001"
                    },
                    {
                        "PreferredAvailabilityZone": "us-east-1d",
                        "CacheNodeId": "0001",
                        "CacheClusterId": "my-redis-cluster-0001-002"
                    }
                ]
            }
        ],
        "ConfigurationEndpoint": {
            "Port": 6379,
            "Address": "my-redis-
cluster.r7gdfi.clustercfg.use1.cache.amazonaws.com"
        },
        "ClusterEnabled": true,
        "ReplicationGroupId": "my-redis-cluster",
        "SnapshotRetentionLimit": 1,
        "AutomaticFailover": "enabled",
        "SnapshotWindow": "07:30-08:30",
        "MemberClusters": [
            "my-redis-cluster-0001-001",
            "my-redis-cluster-0001-002"
        ],
        "CacheNodeType": "cache.t2.micro",
         "DataTiering": "disabled"
        "PendingModifiedValues": {}
    }
```

```
}
```

Para obtener más información consulte modify-replication-group en la Referencia de la AWS CLI.

3. Si ha utilizado `--apply-immediately`, compruebe el estado del clúster utilizando el comando `describe-cache-clusters` de la AWS CLIcon el siguiente parámetro. Cuando el estado cambie a disponible, puede comenzar con el nuevo nodo de clúster de menor tamaño.

Reducción vertical de clústeres de Valkey o Redis OSS (API de ElastiCache)

El siguiente proceso escala su grupo de reproducción de su tipo de nodo actual a un nuevo tipo de nodo más pequeño mediante la API de ElastiCache. Durante este proceso, el clúster de Valkey o Redis OSS seguirá atendiendo solicitudes con un tiempo de inactividad mínimo.

El tiempo que se tarda en la realización del escalado descendente a un tipo de nodo más pequeño varía en función de su tipo de nodo y de la cantidad de datos de su clúster actual.

Reducción vertical (API de ElastiCache)

1. Determine a qué tipos de nodos puede reducir verticalmente mediante la acción `ListAllowedNodeTypeModifications` de la API de ElastiCache con el siguiente parámetro.

   - `ReplicationGroupId`: nombre del grupo de replicación. Use este parámetro para describir un grupo de replicación específico en lugar de todos los grupos de replicación.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=ListAllowedNodeTypeModifications
    &ReplicationGroupId=MyReplGroup
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
    &X-Amz-Credential=<credential>
```

Para obtener más información, consulte ListAllowedNodeTypeModifications en la Referencia de la API de Amazon ElastiCache.

2. Aplique el escalado horizontal a su grupo de reproducción actual al nuevo tipo de nodo mediante la acción `ModifyReplicationGroup` de la API de ElastiCache y con los siguientes parámetros.

   - `ReplicationGroupId`: nombre del grupo de replicación.
   - `CacheNodeType`: nuevo tipo de nodo más pequeño de los clústeres en este grupo de replicación. Este valor debe ser uno de los tipos de instancia devueltos por la acción `ListAllowedNodeTypeModifications` en el paso anterior.
   - `CacheParameterGroupName`: [Opcional] Utilice este parámetro si va a utilizar `reserved-memory` para administrar la memoria reservada de su clúster. Especifique un grupo de parámetros de caché personalizado que reserve la cantidad correcta de memoria para el nuevo tipo de nodo. Si va a utilizar `reserved-memory-percent`, puede omitir este parámetro.
   - `ApplyImmediately`: establézcalo en `true` para que el proceso de reducción vertical se aplique de inmediato. Para aplazar el proceso de escalado descendente al siguiente periodo de mantenimiento, utilice `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=ModifyReplicationGroup
    &ApplyImmediately=true
    &CacheNodeType=cache.t2.micro
    &CacheParameterGroupName=redis32-m3-2xl
    &ReplicationGroupId=myReplGroup
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &Version=2014-12-01
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20141201T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Para obtener más información, consulte ModifyReplicationGroup en la Referencia de la API de Amazon ElastiCache.

Introducción a los filtros de Bloom

ElastiCache admite la estructura de datos de filtros de Bloom, que es una estructura de datos de probabilidad y eficiente en términos de espacio que se utiliza para comprobar si un elemento es miembro de un conjunto. Cuando se utilizan filtros de Bloom, es posible que se generen falsos positivos: un filtro puede indicar incorrectamente la existencia de un elemento, aunque ese elemento no se haya añadido al conjunto. Sin embargo, el uso de los filtros de Bloom evitará los falsos negativos, es decir, indicaciones incorrectas de que un elemento no existe aunque se haya añadido al conjunto.

Puede establecer el porcentaje de posibles falsos positivos en la ratio que prefiera para la carga de trabajo ajustando la ratio de fp. También puede configurar la capacidad (la cantidad de elementos que puede contener un filtro de Bloom), las propiedades escalables y no escalables, y mucho más.

Después de crear un clúster con una versión de motor compatible, el tipo de datos Bloom y los comandos asociados están disponibles de forma automática. El tipo de datos `bloom` es compatible con las API con la sintaxis de comandos del filtro de Bloom de las bibliotecas de cliente oficiales de Valkey, incluidas `valkey-py`, `valkey-java` y `valkey-go`. Puede migrar fácilmente las aplicaciones de Valkey y Redis OSS basadas en Bloom existentes a ElastiCache. Para obtener una lista completa de los comandos, consulte Comandos de filtros de Bloom.

Las métricas relacionadas con Bloom `BloomFilterBasedCmds`, `BloomFilterBasedCmdsLatency` y `BloomFilterBasedCmdsECPUs` se incorporan a CloudWatch para supervisar el uso de este tipo de datos. Para obtener más información, consulte Métricas de Valkey y Redis OSS.

> **Note**
>
> - Para usar los filtros de Bloom, debe ejecutar ElastiCache Valkey 8.1 y versiones posteriores.
> - El tipo de datos Bloom no es compatible con RDB con otras ofertas de Bloom que no estén basadas en Valkey.

Información general del tipo de datos Bloom

Los filtros de Bloom son una estructura de datos probabilística que ahorra espacio y que permite añadir elementos y comprobar si existen elementos. Pueden darse falsos positivos si un filtro indica

incorrectamente que un elemento existe, aunque no se haya añadido. Sin embargo, los filtros de Bloom garantizan que no se produzcan falsos negativos (lo que indica incorrectamente que un elemento no existe aunque se haya añadido).

La principal fuente de documentación de los filtros de Bloom está en la página de documentación de valkey.io. Aquí se incluye la siguiente información:

- Casos de uso habituales de los filtros de Bloom
  - Desduplicación de anuncios o eventos
  - Detección de fraudes
  - Filtrado de contenido nocivo o spam
  - Detección única de usuarios
- Diferencias entre los filtros de Bloom escalables y no escalables
  - Cómo decidirse entre los filtros de Bloom escalables y no escalables
- Propiedades de Bloom
  - Obtenga información sobre las propiedades ajustables de los filtros de Bloom. Esto incluye la ratio de falsos positivos, la capacidad, las propiedades de escalado y no escalado, y mucho más.
- Rendimiento de los comandos de Bloom
- Supervisión de las estadísticas generales de los filtros de Bloom
- Gestión de filtros de Bloom de gran tamaño
  - Recomendaciones y detalles sobre cómo comprobar si un filtro de Bloom está alcanzando su límite de uso de memoria y si puede escalarse para alcanzar la capacidad deseada.
  - Puede comprobar la cantidad de memoria que consume un documento de filtro de Bloom mediante el uso del comando BF.INFO.

Límite del tamaño de Bloom

El consumo de memoria de un único objeto de filtro de Bloom está limitado a 128 MB. Puede comprobar la cantidad de memoria que consume un filtro de Bloom mediante el uso del comando `BF.INFO <key> SIZE`.

ACL de Bloom

De forma similar a las categorías existentes por tipo de datos (@string, @hash, etc.), se añade una nueva categoría @bloom para simplificar la administración del acceso a los comandos y datos de Bloom. Ningún otro comando de Valkey o Redis OSS existente es miembro de la categoría @bloom.

Existen tres categorías de ACL que se actualizan para incluir los nuevos comandos de Bloom: @read, @write y @fast. La siguiente tabla indica la asignación de los comandos de Bloom a las categorías apropiadas.

Comando de Bloom	@read	@write	@fast	@bloom
BF.ADD		y	y	y
BF.CARD	y		y	y
BF.EXISTS	y		y	y
BF.INFO	y		y	y
BF.INSERT		y	y	y
BF.MADD		y	y	y
BF.MEXISTS	y		y	y
BF.RESERVE	y		y	y

Métricas relacionadas con el filtro de Bloom

Se proporcionan las siguientes métricas de CloudWatch relacionadas con las estructuras de datos de Bloom:

Métricas de CW	Unidad	Sin servidor/basado en nodos	Descripción
BloomFilterBasedCmds	Recuento	Ambos	Número total de comandos de

Métricas de CW	Unidad	Sin servidor/basado en nodos	Descripción
			Bloom, incluidos los comandos de lectura y escritura.
BloomFilterBasedCmdsLatency	Microsegundos	Autoadministrado	Latencia de todos los comandos de filtros de Bloom, incluidos los comandos de lectura y escritura.
BloomFilterBasedCmdsECPUs	Recuento	Sin servidor	ECPU consumidas por todos los comandos de filtros de Bloom, incluidos los comandos de lectura y escritura.

Comandos de filtros de Bloom

Los comandos de filtros de Bloom se documentan en el sitio web Valkey.io. Cada página de comandos proporciona una descripción completa de los comandos de Bloom, incluida su sintaxis, comportamiento, valores de retorno y posibles condiciones de error.

Nombre	Descripción
BF.ADD	Añade un único elemento a un filtro de Bloom. Si el filtro aún no existe, se crea.
BF.CARD	Devuelve la cardinalidad de un filtro de Bloom.
BF.EXISTS	Determina si el filtro de Bloom contiene el elemento especificado.
BF.INFO	Devuelve la información de uso y las propiedades de un filtro de Bloom específico.

Nombre	Descripción
BF.INSERT	Crea un filtro de Bloom con cero o más elementos, o añade elementos a un filtro de Bloom existente.
BF.MADD	Añade uno o más elementos a un filtro de Bloom.
BF.MEXISTS	Determina si el filtro de Bloom contiene uno o más elementos.
BF.RESERVE	Crea un filtro de Bloom vacío con las propiedad es especificadas.

> **Note**
>
> ElastiCache no admite BF.LOAD. Solo es relevante para el uso de AOF, que ElastiCache no admite.

Introducción a la búsqueda vectorial

Amazon ElastiCache para Valkey admite la búsqueda vectorial, que le permite almacenar, buscar y actualizar miles de millones de incrustaciones vectoriales de alta dimensión en la memoria con latencias tan bajas como de microsegundos y recuperaciones superiores al 99 %. ElastiCache para Valkey ofrece funciones para indexar, buscar y actualizar miles de millones de incrustaciones vectoriales de alta dimensión de proveedores populares como Amazon Bedrock, Amazon SageMaker, Anthropic u OpenAI para una búsqueda y recuperación rápidas. La búsqueda vectorial de Amazon ElastiCache es ideal para casos de uso en los que el rendimiento máximo y la escalabilidad son los criterios de selección más importantes. Esto incluye el almacenamiento en caché semántico, la generación aumentada de recuperación, las recomendaciones en tiempo real, la personalización y la detección de anomalías.

La búsqueda vectorial se puede utilizar junto con otras característica de ElastiCache para mejorar las aplicaciones. La búsqueda vectorial para ElastiCache está disponible con la versión 8.2 de Valkey en clústeres basados en nodos de todas las regiones de AWS sin costo adicional. Para empezar, cree

un nuevo clúster de Valkey 8.2 con Consola de administración de AWS, el AWS SDK o la AWS CLI. También puede utilizar la búsqueda vectorial en el clúster existente actualizándolo desde cualquier versión de Valkey, o Redis OSS, a Valkey 8.2, con unos pocos clics y sin tiempo de inactividad.

Información general de la búsqueda vectorial

ElastiCache para Valkey ofrece funciones para indexar, buscar y actualizar miles de millones de incrustaciones vectoriales de alta dimensión. La búsqueda vectorial le permite crear, mantener y usar índices secundarios para una búsqueda eficiente y escalable. Cada operación de búsqueda vectorial se aplica a un único índice. Las operaciones de índice se aplican únicamente al índice especificado. Se puede realizar cualquier número de operaciones con cualquier índice en cualquier momento, a excepción de las operaciones de creación y eliminación de índices. En el clúster es posible que se realicen simultáneamente varias operaciones con varios índices.

En este documento, los términos clave, fila y registro son idénticos y se usan indistintamente. Del mismo modo, los términos columna, campo, ruta y miembro también se usan indistintamente.

El comando `FT.CREATE` se puede utilizar para crear un índice para un subconjunto de claves con los tipos de índice especificados. `FT.SEARCH` realiza consultas en los índices creados y `FT.DROPINDEX` elimina un índice existente y todos los datos asociados. No existen comandos especiales para añadir, eliminar o modificar los datos indexados. Los comandos `HASH` o `JSON` existentes que modifican una clave que está en un índice también lo actualizan automáticamente.

Temas

- Los índices y el espacio de claves de Valkey OSS
- El campo de índice escribe
- Algoritmos de índice vectorial
- Seguridad de búsqueda vectorial

Los índices y el espacio de claves de Valkey OSS

Los índices se construyen y mantienen en un subconjunto del espacio de claves de Valkey OSS. Durante la creación del índice se proporciona una lista de prefijos clave que definen el espacio de claves de cada índice. La lista de prefijos es opcional y, si se omite, todo el espacio de claves formará parte de ese índice. Los índices múltiples pueden elegir subconjuntos disociados o superpuestos del espacio de claves sin limitación alguna.

Los índices también están tipificados en el sentido de que solo incluyen las claves de tipo coincidente. Actualmente, solo se admiten los índices de los tipos JSON y HASH. Un índice HASH solo indexa las claves HASH incluidas en su lista de prefijos y, de manera semejante, un índice JSON solo indexa las claves JSON incluidas en su lista de prefijos. Las claves incluidas en la lista de prefijos del espacio de claves de un índice que no poseen el tipo designado se ignoran y no afectan a las operaciones de búsqueda.

Cuando un comando modifica una clave que se encuentra dentro del espacio de claves de un índice, dicho índice se actualiza. Valkey extrae automáticamente los campos declarados para cada índice y actualiza el índice con el nuevo valor. El proceso de actualización consta de tres pasos. En el primer paso, se modifica la clave HASH o JSON y se bloquea el cliente solicitante. El segundo paso se realiza en segundo plano y actualiza cada uno de los índices que contienen la clave modificada. En el tercer paso, se desbloquea el cliente. Por lo tanto, en el caso de las operaciones de consulta realizadas en la misma conexión que una mutación, ese cambio es visible inmediatamente en los resultados de la búsqueda.

La creación de un índice es un proceso de varios pasos. El primer paso es ejecutar el comando FT.CREATE que define el índice. Al ejecutarse correctamente el comando create, se inicia automáticamente el segundo paso: la reposición. El proceso de reposición se ejecuta en un subproceso en segundo plano y analiza el espacio de claves en busca de claves que estén dentro de la lista de prefijos del nuevo índice. Cada clave que se encuentra se agrega al índice. Finalmente, se analiza todo el espacio de claves y se completa el proceso de creación del índice. Tenga en cuenta que mientras el proceso de relleno está en marcha, se permiten las mutaciones de las claves indexadas, no hay restricciones y el proceso de relleno del índice no finalizará hasta que todas las claves estén indexadas correctamente. No se permiten las operaciones de consulta realizadas mientras se está rellenando un índice y se las finaliza con un error. El comando FT.INFO devuelve el estado del proceso de relleno en el campo “backfill_status”.

El campo de índice escribe

Cada índice tiene un tipo específico que se declara cuando se crea el índice junto con la ubicación dentro de un campo (columna) que se va a indexar. En claves HASH, la ubicación es el nombre del campo dentro del HASH. En claves JSON, la ubicación es una descripción de la ruta JSON. Al modificar una clave, los datos asociados a los campos declarados se extraen, se convierten al tipo declarado y se almacenan en el índice. Si faltan los datos o no se pueden convertir correctamente al tipo declarado, ese campo se omite del índice. Hay cuatro tipos de campo, según se explica a continuación:

- Los campos vectoriales contienen un vector de números, también conocido como incrustación de vectores. Los campos vectoriales se pueden usar para filtrar vectores en función de métricas de distancia específicas que miden la similitud. En índices HASH, el campo debe contener todo el vector codificado en formato binario (IEEE 754 del tipo little-endian). En claves JSON, la ruta debe hacer referencia a una matriz del tamaño correcto llena de números. Tenga en cuenta que cuando se utiliza una matriz JSON como campo vectorial, la representación interna de la matriz dentro de la clave JSON se convierte al formato exigido por el algoritmo seleccionado, lo que reduce el consumo y la precisión de memoria. Las operaciones de lectura posteriores con los comandos JSON darán como resultado el valor de precisión reducido.
- Los campos numéricos contienen un solo número. Los campos numéricos se pueden utilizar con el operador de búsqueda por rangos. En HASH, se espera que el campo contenga el texto ASCII de un número escrito en el formato estándar para números de punto fijo o flotante. En campos JSON, se deben seguir las reglas numéricas de los números JSON. Independientemente de la representación que contenga la clave, este campo se convierte en un número de punto flotante de 64 bits para almacenarlo en el índice. Como los números subyacentes se almacenan en punto flotante con sus limitaciones de precisión, se aplican las reglas habituales sobre las comparaciones numéricas de números de punto flotante.
- Los campos de etiquetas contienen cero o más valores de etiqueta codificados como una sola cadena UTF-8. Los campos de etiquetas se pueden usar para filtrar las consultas y determinar la equivalencia de los valores de las etiquetas mediante una comparación que distinga entre mayúsculas y minúsculas o que no distinga entre mayúsculas y minúsculas. La cadena se analiza en valores de etiqueta mediante un carácter separador (el valor predeterminado es una coma, pero se puede anular) y se eliminan los espacios en blanco iniciales y finales. Se puede incluir cualquier número de valores de etiqueta en un único campo de etiqueta.

Algoritmos de índice vectorial

Valkey admite dos algoritmos de índice vectorial:

- FLAT: el algoritmo Flat es un procesamiento lineal de fuerza bruta de cada vector del índice, que da como resultado respuestas exactas dentro de los límites de la precisión de los cálculos de distancia. Debido al procesamiento lineal del índice, los tiempos de ejecución de este algoritmo pueden ser muy altos para índices grandes. Los índices planos admiten velocidades de ingesta más altas.
- Pequeño mundo navegable jerárquico (HNSW): el algoritmo HNSW es una alternativa que proporciona una aproximación de las coincidencias vectoriales más cercanas a cambio de tiempos

de ejecución considerablemente más bajos. El algoritmo está controlado por tres parámetros, `M` `EF_CONSTRUCTION` y `EF_RUNTIME`. Los dos primeros parámetros se especifican en el momento de la creación del índice y no se pueden cambiar. El parámetro `EF_RUNTIME` tiene un valor predeterminado que se especifica al crear el índice, pero se puede anular posteriormente en cualquier operación de consulta individual. Estos tres parámetros interactúan para equilibrar el consumo de memoria y de CPU durante las operaciones de incorporación y consulta, así como para controlar la calidad de la aproximación de una búsqueda KNN exacta (conocida como relación de recuperación).

En HNSW, el parámetro M controla el número máximo de vecinos a los que se puede conectar cada nodo, lo que da forma a la densidad del índice. Una M más alta, de 32 o superior, produce un gráfico más conectado, lo que mejora la velocidad de recuperación y consulta, ya que existen más rutas para llegar a los vecinos pertinentes. Sin embargo, aumenta el tamaño del índice y del uso de memoria y ralentiza la indexación. Una M más baja, de 8 o inferior, produce un índice más pequeño, más rápido de crear y con un menor uso de memoria, pero la recuperación disminuye y las consultas pueden tardar más debido al menor número de conexiones.

El parámetro EF_construction indica cuántas conexiones candidatas se evalúan al crear el índice. Un EF_construction más alto, de 400 o superior, significa que el indexador considera más rutas antes de seleccionar los vecinos, lo que se traduce en un gráfico que mejora la eficiencia tanto de la recuperación como de las consultas en el futuro, pero a costa de una indexación más lenta y un mayor uso de CPU y memoria durante la construcción. Un EF_construction bajo, de entre 64 y 120, acelera la indexación y reduce el uso de recursos, pero el gráfico resultante puede reducir la recuperación y la lentitud de las consultas, incluso si EF_runtime tiene un valor alto.

Por último, EF_runtime regula la amplitud de la búsqueda durante la consulta, controlando el número de vecinos candidatos que se explora en tiempo de ejecución. Si se establece en un valor alto, aumenta la recuperación y la precisión, pero a costa de la latencia de las consultas y del uso de la CPU. Un EF_runtime bajo hace que las consultas sean más rápidas y ligeras, pero con una menor capacidad de recuperación. A diferencia de M o EF_construction, este parámetro no afecta al tamaño del índice ni al tiempo de creación, por lo que se convierte en el parámetro para ajustar las compensaciones entre recuperación y latencia una vez creado un índice.

Ambos algoritmos de búsqueda vectorial (FLAT y HNSW) admiten un parámetro `INITIAL_CAP` opcional. Si se especifica, este parámetro asigna previamente memoria a los índices, lo que da como resultado una reducción de la sobrecarga de administración de la memoria y aumenta las tasas de incorporación vectorial. Los índices planos admiten mejores velocidades de ingesta que HNSW.

Es posible que los algoritmos de búsqueda vectorial, como el HNSW, no gestionen de manera eficiente la eliminación o la sobrescritura de los vectores previamente insertados. El uso de estas operaciones puede provocar un consumo excesivo de memoria indexada o una calidad de la recuperación reducida. La reindexación es un método para restablecer el uso o la recuperación óptimos de la memoria.

Seguridad de búsqueda vectorial

Los mecanismos de seguridad de ACL (listas de control de acceso) de Valkey para el acceso a los comandos y a los datos se han ampliado para controlar la función de búsqueda. El control ACL de los comandos de búsqueda individuales es totalmente compatible. Se proporciona una nueva categoría de ACL, `@search`, y muchas de las categorías existentes (`@fast`, `@read`, `@write`, etc.) se actualizan para incluir los nuevos comandos. Los comandos de búsqueda no modifican los datos clave, lo que significa que se conserva la maquinaria ACL existente para el acceso de escritura. La presencia de un índice no modifica las reglas de acceso para las operaciones `HASH` y `JSON`; se sigue aplicando el control de acceso normal en la clave a estos comandos.

El acceso de los comandos de búsqueda con un índice también se controla mediante la ACL. Las comprobaciones de acceso se realizan a nivel de índice completo, no al nivel de la clave. Esto significa que el acceso a un índice se garantiza a un usuario solo si ese usuario tiene permiso para acceder a todas las claves posibles de la lista de prefijos del espacio de claves de ese índice. En otras palabras, el contenido real de un índice no controla el acceso. Más bien, es el contenido teórico de un índice, tal como se define en la lista de prefijos, el que se utiliza para el control de seguridad. Se pueden generar situaciones en las que un usuario tiene acceso de lectura o escritura a una clave, pero no tiene acceso a un índice que contenga esa clave. Tenga en cuenta que solo se requiere acceso de lectura al espacio de claves para crear o usar un índice; no se tiene en cuenta la presencia o ausencia del acceso de escritura.

Características y límites de la búsqueda vectorial

Disponibilidad de búsqueda vectorial

La búsqueda vectorial para Amazon ElastiCache está disponible con la versión 8.2 de Valkey en clústeres basados en nodos de todas las regiones de AWS sin costo adicional. También puede utilizar la búsqueda vectorial en los clústeres existentes actualizándolos desde cualquier versión de Valkey o Redis OSS a Valkey 8.2 con unos pocos clics y sin tiempo de inactividad.

La búsqueda vectorial está disponible en todos los tipos de instancia compatibles con ElastiCache. El uso de la búsqueda vectorial en las instancias t2, t3 y t4g requiere aumentar la reserva de memoria

al menos al 50 % para las instancias micro y al 30 % para las instancias pequeñas. Consulte esta página para obtener más información.

Restricciones paramétricas

En la siguiente tabla se muestran los límites de varios elementos de búsqueda vectorial:

Límites de la búsqueda vectorial

Elemento	Valor máximo
Cantidad de dimensiones de un vector	32 768
Cantidad de índices que se pueden crear	10
Cantidad de campos de un índice	50
Cláusula FT.SEARCH TIMEOUT (milisegundos)	60 000
Número máximo de prefijos permitido por índice	16
Longitud máxima de un campo de etiqueta	10000
Longitud máxima de un campo numérico	256
Parámetro HNSW M	2000000
Parámetro HNSW EF_CONSTRUCTION	4096
Parámetro HNSW EF_RUNTIME	4096

Restricciones operativas

Persistencia y reposición de índices

El proceso de actualización consta de tres pasos. En el primer paso, se modifica la clave HASH o JSON y se bloquea el cliente solicitante. El segundo paso se realiza en segundo plano y actualiza cada uno de los índices que contienen la clave modificada. En el tercer paso, se desbloquea el cliente. Por lo tanto, en el caso de las operaciones de consulta realizadas en la misma conexión

que una mutación, ese cambio es visible inmediatamente en los resultados de la búsqueda. Por lo tanto, la inserción o actualización de una clave no será visible en los resultados de búsqueda para otros clientes durante un breve período de tiempo. Durante los períodos en los que el sistema está sobrecargado o hay grandes cambios en los datos, el retraso en la visibilidad puede prolongarse.

La característica de búsqueda vectorial conserva la definición de los índices y el contenido de los índices. Los índices de los campos vectoriales se guardan, pero los índices de TAGS y NUMERIC no, por lo que deben reconstruirse cuando se cargan externamente (sincronización completa o recarga). Esto significa que, durante cualquier solicitud o evento operativo que provoque el inicio o el reinicio de un nodo, la definición y el contenido del índice para los vectores se restauran a partir de la última instantánea. Para iniciar esta operación no se requiere ninguna acción por parte del usuario. Sin embargo, para los índices TAGS y NUMERIC, la reconstrucción se realiza como una operación de rellenado en cuanto se restauran los datos. Esto equivale funcionalmente a que el sistema ejecute automáticamente un comando FT.CREATE para cada índice definido. Tenga en cuenta que el nodo estará disponible para las operaciones de la aplicación tan pronto como se restauren los datos, pero probablemente antes de que se complete el rellenado del índice, lo que significa que las operaciones de rellenado volverán a estar visibles para las aplicaciones.

La finalización de la reposición del índice no se sincroniza entre una reposición principal y una réplica. Esta falta de sincronización puede pasar a ser visible para las aplicaciones de forma inesperada, por lo que se recomienda que las aplicaciones verifiquen que esté finalizada el relleno en las principales y todas las réplicas antes de iniciar las operaciones de búsqueda.

Límites de escalado

Durante los eventos de escalado, es posible que el índice se rellene a medida que se migran los datos. Esto se traduce en una menor recuperación de las consultas de búsqueda.

Importación y exportación de instantáneas y migración en tiempo real

Los archivos RDB de un clúster con índices de búsqueda se pueden importar a otro clúster de ElastiCache Valkey con la versión 8.2 o superior. El nuevo clúster reconstruirá el contenido del índice al cargar el archivo RDB. Sin embargo, la presencia de índices de búsqueda en un archivo RDB limita la compatibilidad de esos datos con las versiones anteriores de Valkey. El formato de los índices de búsqueda definido por la funcionalidad de búsqueda vectorial solo lo entiende otro clúster de ElastiCache con Valkey versión 8.2 o superior. Sin embargo, los archivos RDB que no contienen índices no están restringidos de esta manera.

Falta de memoria durante la reposición

Al igual que en las operaciones de escritura de Valkey OSS, el relleno de índices está sujeto a limitaciones por falta de memoria. Si la memoria del motor se llena mientras hay una reposición en curso, todas las reposiciones se pausan. Si queda memoria disponible, se reanuda el proceso de reposición. También se puede eliminar un índice cuando el relleno está en pausa por falta de memoria.

Transacciones

Los comandos `FT.CREATE`, `FT.DROPINDEX`, `FT.ALIASADD`, `FT.ALIASDEL` y `FT.ALIASUPDATE` no se pueden ejecutar en un contexto transaccional, es decir, no dentro de un bloque `MULTI/EXEC` ni dentro de un script LUA o FUNCTION.

Selección de la configuración adecuada

En la experiencia con la consola, ElastiCache ofrece una forma sencilla de elegir el tipo de instancia adecuado en función de los requisitos de memoria y CPU de la carga de trabajo vectorial.

Consumo de memoria

El consumo de memoria se basa en el número de vectores, el número de dimensiones, el valor M y la cantidad de datos no vectoriales, como los metadatos asociados al vector u otros datos almacenados en la instancia. La memoria total necesaria es una combinación del espacio necesario para los datos vectoriales reales y el espacio necesario para los índices vectoriales. El espacio necesario para los datos vectoriales se calcula midiendo la capacidad real necesaria para almacenar los vectores dentro de las estructuras de datos `HASH` o `JSON` y la sobrecarga a los slabs de memoria más cercanos, para lograr asignaciones de memoria óptimas. Cada uno de los índices vectoriales utiliza referencias a los datos vectoriales almacenados en estas estructuras de datos, así como una copia adicional del vector del índice. Se recomienda planificar este consumo de espacio adicional mediante el índice.

El número de vectores depende de cómo se decida representar los datos como vectores. Por ejemplo, puede elegir representar un único documento en varios fragmentos, donde cada fragmento represente un vector. Como alternativa, puede optar por representar todo el documento como un único vector. El número de dimensiones de los vectores depende del modelo de incrustación que se elija. Por ejemplo, si opta por utilizar el modelo de incrustación AWSTitan, el número de dimensiones sería 1536. Tenga en cuenta que debe probar el tipo de instancia para asegurarse de que se ajuste a sus requisitos.

Escalado de la carga de trabajo

La búsqueda vectorial admite tres métodos de escalado: horizontal, vertical y réplicas. Al escalar por capacidad, la búsqueda vectorial se comporta igual que en Valkey, es decir, si se aumenta la memoria de los nodos individuales (escalado vertical) o se incrementa el número de nodos (escalado horizontal) crece la capacidad total. En el modo de clúster, el comando `FT.CREATE` se puede enviar a cualquier nodo principal del clúster y el sistema distribuirá automáticamente la nueva definición del índice a todos los miembros del clúster.

Sin embargo, desde el punto de vista del rendimiento, la búsqueda vectorial se comporta de forma muy diferente a la de Valkey normal. La implementación de varios subprocesos de la búsqueda vectorial significa que las CPU adicionales producen aumentos lineales tanto en el rendimiento de las consultas como de la ingesta. El escalado horizontal produce aumentos lineales en el rendimiento de la ingesta, pero podría reducir el rendimiento de las consultas. Si se requiere un rendimiento de las consultas adicional, es necesario escalar mediante réplicas o CPU adicionales.

Comandos de búsqueda vectorial

A continuación se muestra una lista de comandos compatibles para la búsqueda vectorial.

Temas

- FT.CREATE
- FT.SEARCH
- FT.DROPINDEX
- FT.INFO
- FT._LIST

FT.CREATE

El comando `FT.CREATE` crea un índice vacío e inicia el proceso de relleno. Cada índice consta de una serie de definiciones de campo. Cada definición de campo especifica un nombre de campo, un tipo de campo y una ruta dentro de cada clave indexada para localizar un valor del tipo declarado. Algunas definiciones de tipos de campo tienen especificadores de subtipos adicionales.

Para los índices de las claves HASH, la ruta es la misma que el nombre del miembro hash. Si lo desea, puede utilizar la cláusula `AS` opcional para cambiar el nombre del campo. El cambio de nombre de los campos es especialmente útil cuando el nombre del miembro contiene caracteres especiales.

En el caso de los índices de claves JSON, la ruta es una ruta JSON a los datos del tipo declarado. Como la ruta JSON siempre contiene caracteres especiales, la cláusula AS es obligatoria.

Sintaxis

```
FT.CREATE <index-name>
ON HASH | JSON
[PREFIX <count> <prefix1> [<prefix2>...]]
SCHEMA
(<field-identifier> [AS <alias>]
| VECTOR [HNSW|FLAT] <attr_count> [<attribute_name> <attribute_value>])
| TAG [SEPARATOR <sep>] [CASESENSITIVE]
| NUMERIC
)+
```

<index-name**>** (obligatorio): es el nombre que le asigna al índice. Si ya existe un índice con el mismo nombre, se devuelve un error.

ON HASH | JSON (opcional): en este índice solo se incluyen las claves que coinciden con el tipo especificado. Si se omite, se toma HASH.

PREFIX **<**prefix-count**>** **<**prefix**>** (opcional): si se especifica esta cláusula, solo se incluirán en este índice las claves que comiencen con los mismos bytes que uno o más de los prefijos especificados. Si se omite esta cláusula, se incluirán todas las claves del tipo correcto. Un prefijo de longitud cero también coincidiría con todas las claves del tipo correcto.

Tipos de campo:

- TAG: un campo de etiqueta es una cadena que contiene uno o más valores de etiqueta.
  - SEPARATOR **<**sep**>** (opcional): es uno de los caracteres `,.<>{}[]"':;!@#$%^&*()-+=~` que se utiliza para delimitar etiquetas individuales. Si se omite, el valor predeterminado es `,`.
  - CASESENSITIVE (opcional): si está presente, las comparaciones de etiquetas distinguirán entre mayúsculas y minúsculas. De forma predeterminada, las comparaciones de etiquetas NO distinguen entre mayúsculas y minúsculas.
- NUMERIC: un campo numérico contiene un número.
- VECTOR: un campo vectorial contiene un vector. Actualmente se admiten dos algoritmos de indexación vectorial: HNSW (pequeño mundo navegable jerárquico) y FLAT (fuerza bruta). Cada algoritmo tiene un conjunto de atributos adicionales, algunos obligatorios y otros opcionales.

- FLAT: el algoritmo Flat proporciona respuestas exactas, pero tiene un tiempo de ejecución proporcional al número de vectores indexados y, por lo tanto, puede no ser apropiado para conjuntos de datos de gran tamaño.
  - DIM <number> (obligatorio): especifica el número de dimensiones de un vector.
  - TYPE FLOAT32 (obligatorio): tipo de datos; actualmente solo se admite FLOAT32.
  - DISTANCE_METRIC [L2 | IP | COSINE] (obligatorio): especifica el algoritmo de distancia.
  - INITIAL_CAP <size> (opcional): tamaño del índice inicial.
- HNSW: el algoritmo HNSW proporciona respuestas aproximadas, pero funciona mucho más rápido que el FLAT.
  - DIM <number> (obligatorio): especifica el número de dimensiones de un vector.
  - TYPE FLOAT32 (obligatorio): tipo de datos; actualmente solo se admite FLOAT32.
  - DISTANCE_METRIC [L2 | IP | COSINE] (obligatorio): especifica el algoritmo de distancia.
  - INITIAL_CAP <size> (opcional): tamaño del índice inicial.
  - M <number> (opcional): número máximo de periferias salientes permitido para cada nodo del gráfico en cada capa. En la capa cero, el número máximo de periferias salientes será de 2*M. El valor predeterminado es 16 y el máximo es 512.
  - EF_CONSTRUCTION <number> (opcional): controla la cantidad de vectores examinados durante la construcción del índice. Los valores más altos de este parámetro mejorarán la tasa de recuperación a costa de prolongar los tiempos de creación del índice. El valor predeterminado es 200. El valor máximo es 4096.
  - EF_RUNTIME <number> (opcional): controla el número de vectores que se va a examinar durante una operación de consulta. El valor predeterminado es 10 y el máximo es 4096. Puede establecer este valor de parámetro para cada consulta que ejecute. Los valores más altos aumentan los tiempos de consulta, pero mejoran la recuperación de las consultas.

RESPONSE: OK o error.

FT.SEARCH

Realiza una búsqueda en el índice especificado. Se devuelven las claves que coinciden con la expresión de la consulta.

```
FT.SEARCH <index-name> <query>
[NOCONTENT]
[RETURN <token_count> (<field-identifier> [AS <alias>])+]
```

```
[TIMEOUT timeout]
[PARAMS <count> <name> <value> [<name> <value>]]
[LIMIT <offset> <count>]
```

- <index> (obligatorio): es el nombre del índice que desea consultar.
- <query> (obligatorio): es la cadena de consulta. Vea la información que aparece a continuación.
- NOCONTENT (opcional): si está presente, solo se devuelven los nombres de clave resultantes y no se incluyen los valores clave.
- TIMEOUT <timeout> (opcional): permite establecer un valor de tiempo de espera para el comando de búsqueda. Debe ser un número entero en milisegundos.
- PARAMS <count> <name1> <value1> <name2> <value2> ... (opcional): `count` es el número de argumentos, es decir, el doble del número de pares de nombres de valores. Observe la cadena de consulta para obtener información detallada sobre el uso.
- RETURN <count> <field1> <field2> ... (opcional): es el recuento del número de campos que se va a devolver. Especifica los campos que desea recuperar de los documentos, junto con los alias de los valores devueltos. De forma predeterminada, se devuelven todos los campos a menos que se establezca la opción NOCONTENT, en cuyo caso no se devuelve ningún campo. Si el recuento se establece en 0, se comporta igual que NOCONTENT.
- LIMIT: <offset> <count>: permite elegir una parte del resultado. Se omiten las primeras claves <offset> y solo se incluye un máximo de <count> claves. El valor predeterminado es LIMIT 0 10, que devuelve como máximo 10 claves.
- PARAMS: dos veces la cantidad de pares de valores clave. Se puede hacer referencia a los pares clave/valor de los parámetros desde la expresión de consulta. Para obtener más información, consulte Expresión de consulta de búsqueda vectorial.
- DIALECT: <dialect> (opcional): especifica el dialecto. El único dialecto admitido es el 2.

RESPONSE

El comando devuelve una matriz si tiene éxito o se trata de un error.

En caso de éxito, la primera entrada de la matriz de respuesta representa el recuento de claves coincidentes, seguida de una entrada de matriz por cada clave coincidente. Tenga en cuenta que si se especifica la opción `LIMIT`, solo controlará el número de claves devueltas y no afectará al valor de la primera entrada.

Cuando se especifica NOCONTENT, cada entrada de la respuesta solo contiene el nombre clave coincidente. De lo contrario, cada entrada incluye el nombre clave coincidente, seguido de una matriz de los campos devueltos. Los campos de resultados de una clave constan de un conjunto de pares de nombre/valor. El primer par nombre/valor corresponde a la distancia calculada. El nombre de este par se construye a partir del nombre del campo vectorial al que se añade “__” delante y “_score” detrás. El valor es la distancia calculada. Los pares nombre/valor restantes son los miembros y valores de la clave controlados por la cláusula RETURN.

La cadena de consulta se ajusta a la siguiente sintaxis:

```
<filtering>=>[ KNN <K> @<vector_field_name> $<vector_parameter_name> <query-
modifiers> ]
```

Donde:

- <filtering>: es un * o una expresión de filtro. Un asterisco (*) significa que no se filtra y, por lo tanto, se buscan todos los vectores del índice. Se puede proporcionar una expresión de filtro para designar un subconjunto de los vectores que se van a buscar.
- <vector_field_name>: es el nombre de un campo vectorial dentro del índice especificado.
- <K>: es el número de vectores vecinos más cercanos que se va a devolver.
- <vector_parameter_name>: es un nombre PARAM cuyo valor correspondiente proporciona el vector de consulta para el algoritmo KNN. Tenga en cuenta que este parámetro debe codificarse como punto flotante binario IEEE 754 de 32 bits en formato little-endian.
- <query-modifiers>: (opcional) es una lista de pares de palabras clave/valor que modifican esta búsqueda de KNN en particular. Actualmente, se admiten dos palabras clave:
  - EF_RUNTIME: esta palabra clave va acompañada de un valor entero que anula el valor predeterminado de EF_RUNTIME especificado cuando se creó el índice.
  - AS: esta palabra clave va acompañada de un valor de cadena que se convierte en el nombre del campo de puntuación del resultado y anula el algoritmo de generación de nombres de campo de puntuación predeterminado.

Expresión de filtro

Una expresión de filtro se crea como una combinación lógica de los operadores de búsqueda numéricos y de etiquetas incluidos entre paréntesis.

Etiqueta

El operador de búsqueda de etiquetas se especifica con una o más cadenas separadas por el carácter |. Una clave satisfará al operador de búsqueda de etiquetas si el campo indicado contiene alguna de las cadenas especificadas.

```
@<field_name>:{<tag>}
or
@<field_name>:{<tag1> | <tag2>}
or
@<field_name>:{<tag1> | <tag2> | ...}
```

Por ejemplo, la siguiente consulta devolverá los documentos de color azul O negro O verde.

```
@color:{blue | black | green}
```

En otro ejemplo, la siguiente consulta devolverá documentos que contengan “hello world” o “hello universe”.

```
@description:{hello world | hello universe}
```

Rango numérico

El operador de rango numérico permite filtrar las consultas para que solo devuelvan valores que se encuentren entre un valor inicial y un valor final determinados. Se admiten consultas de rango inclusivas y exclusivas. Para realizar comparaciones relacionales sencillas, se puede usar +inf o -inf con una consulta de rango. La sintaxis de una búsqueda por rango es la siguiente:

```
@<field_name>:[ [(] <bound> [(] <bound>]
```

...donde **<bound>** es un número o +inf o -inf. Los límites sin un padre abierto principal son inclusivos, mientras que los límites con el padre abierto principal son exclusivos.

Utilice la siguiente tabla como guía para asignar expresiones matemáticas a las consultas de filtrado:

```
min <= field <= max         @field:[min max]
min < field <= max          @field:[(min max]
min <= field < max            @field:[min (max]
min < field < max            @field:[(min (max]
field >= min                @field:[min +inf]
field > min                    @field:[(min +inf]
field <= max                @field:[-inf max]
field < max                    @field:[-inf (max]
field == val                @field:[val val]
```

Operadores lógicos

Se pueden utilizar varias etiquetas y operadores de búsqueda numérica para crear consultas complejas mediante operadores lógicos.

AND lógico

Para establecer un AND lógico, utilice un espacio entre los predicados. Por ejemplo:

```
query1 query2 query3
```

OR lógico

Para establecer un OR lógico, utilice un espacio entre los predicados. Por ejemplo:

```
query1 | query2 | query3
```

Negación lógica

Se puede negar cualquier consulta anteponiendo el carácter - a cada consulta. Las consultas negativas devuelven todas las entradas que no coinciden con la consulta. Esto también incluye las claves que no tienen el campo.

Por ejemplo, una consulta negativa en @genre:{comedy} devolverá todos los libros que no sean comedia Y todos los libros que no tengan un campo de género.

La siguiente consulta devolverá todos los libros del género “comedy” que no se hayan publicado entre 2015 y 2024 o que no tengan ningún campo de año: @genre:[comedy] -@year:[2015 2024]

Jerarquía de los operadores

Se aplican las reglas típicas de prioridad de los operadores, es decir, NEGATE es la prioridad más alta, seguida de la lógica AND y, a continuación, la lógica OR con la prioridad más baja. Para invalidar las reglas de prioridad predeterminadas, puede utilizar paréntesis.

Ejemplos de operadores lógicos combinados

Los operadores lógicos se pueden combinar para formar expresiones de filtro complejas.

La siguiente consulta devolverá todos los libros del género “comedy” o “horror” (AND) publicados entre 2015 y 2024: `@genre:[comedy|horror] @year:[2015 2024]`.

La siguiente consulta devolverá todos los libros del género “comedy” o “horror” (OR) publicados entre 2015 y 2024: `@genre:[comedy|horror] | @year:[2015 2024]`.

La siguiente consulta devolverá todos los libros que no tengan ningún campo de género o que tengan un campo de género distinto de “comedy” y que se hayan publicado entre 2015 y 2024: `-@genre:[comedy] @year:[2015 2024]`

FT.DROPINDEX

Sintaxis

```
FT.DROPINDEX <index-name>
```

Se elimina el índice especificado. Si el índice no existe, devuelve OK o un error.

- **<index-name>** (obligatorio): el nombre del índice que se va a eliminar.

FT.INFO

Sintaxis

```
FT.INFO <index-name>
```

La búsqueda vectorial amplía el comando FT.INFO con varias secciones adicionales de estadísticas y contadores. Al solicitar la recuperación de la sección SEARCH, se recuperarán todas las estadísticas siguientes:

Nombre de la clave	Tipo de valor	Descripción
index_name	cadena	Nombre del índice
index_options	cadena	Reservado. Actualmente está establecido en 0.
index_definition	matriz	Consulte a continuación la definición de estos elementos de la matriz.
attributes	matriz de información de atributos	Hay un elemento de esta matriz definido para cada atributo; consulte a continuac

Nombre de la clave	Tipo de valor	Descripción
		ión la definición de la informaci ón de atributo.
num_docs	entero	Número de claves que contiene actualmente el índice
num_terms	entero	Reservado. Actualmente está establecido en 0.
record_count	entero	Es la suma del campo “size” de cada atributo.
hash_indexing_failures	entero	Número de veces que un atributo no se ha podido convertir al tipo de atributo declarado. A pesar del nombre, esto también se aplica a las claves JSON.
backfill_in_progress	entero	Si hay un relleno en curso, será un 1; de lo contrario, será un 0.
backfill_percent_complete	float	Estimación de la finalización del relleno, que es un número en fracciones en el rango [0.. 1].
mutation_queue_size	entero	Número de claves en espera para actualizar el índice.
recent_mutations_queue_dela y	entero	Estimación del retraso (en segundos) de la actualización del índice. 0 es el valor si no hay actualizaciones en curso.

Nombre de la clave	Tipo de valor	Descripción
estado	cadena	Estado de relleno: “ready” indica que el relleno se ha completado correctamente. “backfill_in_progress” indica que se está rellenando. “backfill_paused_by_oom” significa que el proceso de rellenado se ha pausado debido a que la memoria es insuficiente. Cuando se resuelva el problema de la falta de memoria, el proceso de relleno continuará.

La estructura index_definition es una matriz de pares clave/valor definidos de la siguiente manera:

Nombre de la clave	Tipo de valor	Descripción
key_type	cadena	La cadena JSON o la cadena HASH.
prefixes	matriz	Cada elemento de la matriz es un prefijo definido para el índice. Si no se ha especific ado ningún prefijo al crear el índice, esta matriz tendrá 0 entradas.
default_score	cadena	Reservado. Actualmente está establecido en 1.

Información de los atributos: la información de los atributos es específica del tipo.

Atributos numéricos:

Clave	Tipo de valor	Descripción
identificador	cadena	Ubicación del atributo dentro de una clave. Nombre del miembro Hash o ruta JSON.
alias	cadena	Nombre del atributo utilizado en las descripciones de las consultas.
tipo	cadena	Es la cadena NUMERIC.
tamaño	entero	Es el número de claves con valores numéricos válidos en este atributo.

Atributos de las etiquetas:

Nombre de la clave	Tipo de valor	Descripción
identificador	cadena	Ubicación del atributo dentro de una clave. Nombre del miembro Hash o ruta JSON.
alias	cadena	Nombre del atributo utilizado en las descripciones de las consultas.
tipo	cadena	Es la cadena TAG.
SEPARATOR	character	Es el carácter separador definido cuando se creó el índice.
CASESENSITIVE	n/a	Esta clave no tiene ningún valor asociado. Solo está

Nombre de la clave	Tipo de valor	Descripción
		presente si el atributo se ha creado con esta opción.
tamaño	entero	Es el número de claves con valores de etiqueta válidos en este atributo.

Atributos de los vectores:

Nombre de la clave	Tipo de valor	Descripción
identificador	cadena	Ubicación del atributo dentro de una clave. Nombre del miembro Hash o ruta JSON.
alias	cadena	Nombre del atributo utilizado en las descripciones de las consultas.
tipo	cadena	Es la cadena VECTOR.
índice	character	Para obtener una descripci ón más detallada del índice vectorial, consulte lo que sigue.

Descripción del índice vectorial:

Nombre de la clave	Tipo de valor	Descripción
capacity	cadena	Es la capacidad actual del índice.
dimensiones	cadena	Es el número de elementos de cada vector.

Nombre de la clave	Tipo de valor	Descripción
distance_metric	cadena	Una de las siguientes: COSINE, L2 o IP
tamaño	matriz	Para obtener una descripci ón más detallada del índice vectorial, consulte lo que sigue.
data_type	cadena	Tipo de datos declarado. En este momento solo se admite FLOAT32.
algoritmo	matriz	Descripción más detallada del algoritmo de búsqueda vectorial.

Descripción del algoritmo de búsqueda vectorial FLAT:

Nombre de la clave	Tipo de valor	Descripción
name	cadena	Nombre del algoritmo: FLAT
block_size	número	Tamaño de un bloque del índice FLAT.

Descripción del índice vectorial HNSW:

Nombre de la clave	Tipo de valor	Descripción
name	cadena	Nombre del algoritmo: HNSW
m	número	Parámetro M para HNSW
ef_construction	número	Parámetro “ef_construction” para HNSW

Nombre de la clave	Tipo de valor	Descripción
ef_runtime	número	Parámetro “ef_runtime” para HNSW

FT._LIST

Enumera todos los índices.

Sintaxis

```
FT._LIST
```

Devuelve una matriz de cadenas que son los nombres del índice definido actualmente.

Introducción a JSON para Valkey y Redis OSS

ElastiCache admite el formato nativo JavaScript Object Notation (JSON), que es una forma sencilla y sin esquemas de codificar conjuntos de datos complejos dentro de clústeres de Valkey y Redis OSS. Puede almacenar datos de forma nativa y acceder a ellos utilizando el formato JavaScript Object Notation (JSON) dentro de clústeres, y actualizar los datos de JSON almacenados en esos clústeres; sin necesidad de administrar un código personalizado para serializarlo y deserializarlo.

Además de usar operaciones de API de Valkey y Redis OSS para las aplicaciones que funcionan sobre JSON, ahora puede recuperar y actualizar de manera eficiente partes específicas de un documento de JSON sin necesidad de manipular todo el objeto. Esto puede mejorar el rendimiento y reducir los costos. También puede buscar en el contenido de su documento JSON mediante la consulta Estilo Goessner`JSONPath`.

Después de crear un clúster con una versión de motor compatible, el tipo de datos JSON y los comandos asociados están disponibles automáticamente. API compatible y RDB compatible con la versión 2 del módulo de JSON, por lo que puede migrar fácilmente las aplicaciones de Valkey y Redis OSS basadas en JSON existentes a ElastiCache. Para obtener más información acerca de los comandos admitidos, consulte Comandos de Valkey y Redis OSS compatibles.

Las métricas relacionadas con JSON `JsonBasedCmds` y `JsonBasedCmdsLatency` se incorporan a CloudWatch para supervisar el uso de este tipo de datos. Para obtener más información, consulte Métricas de Valkey y Redis OSS.

> **Note**
>
> Para utilizar JSON, debe ejecutar Valkey 7.2 y versiones posteriores o Redis OSS 6.2.6 o versiones posteriores.

Temas

- Información general del tipo de datos JSON
- Comandos de Valkey y Redis OSS compatibles

Información general del tipo de datos JSON

ElastiCache admite una serie de comandos de Valkey y Redis OSS para trabajar con el tipo de datos JSON. A continuación, se presenta información general del tipo de datos JSON y una lista detallada de los comandos admitidos.

Terminología

Plazo	Descripción
Documento JSON	Hace referencia al valor de una clave de JSON.
Valor JSON	Hace referencia a un subconjunto de un JSON, incluida la raíz que representa a todo el documento. Un valor podría ser un contenedor o una entrada dentro de un contenedor.
Elemento JSON	Equivalente al valor JSON.

Estándares JSON admitidos

El formato JSON es compatible con el estándar de intercambio de datos JSON RFC 7159 y ECMA-404. Se admite UTF-8 Unicode en texto JSON.

Elemento raíz

El elemento raíz puede ser de cualquier tipos de datos de JSON. Tenga en cuenta que en la RFC 4627 anterior, solo se permitían objetos o matrices como valores raíz. Desde la actualización a RFC 7159, la raíz de un documento JSON puede ser de cualquier tipo de datos JSON.

Límite de tamaño del documento

Los documentos JSON se almacenan de manera interna en un formato que se optimiza para lograr un acceso y modificación rápidos. Este formato suele consumir algo más de memoria que la representación serializada equivalente del mismo documento.

El consumo de memoria de un solo documento JSON está limitado a 64 MB, que es el tamaño de la estructura de datos en memoria, no la cadena JSON. Puede comprobar la cantidad de memoria que consume un documento JSON mediante el uso del comando `JSON.DEBUG MEMORY`.

JSON ACL

- Similar a las categorías existentes por tipo de datos (@string, @hash, etc.), se agrega una nueva categoría @json para simplificar la administración del acceso a los comandos y datos JSON. Ningún otro comando de Valkey o Redis OSS existente es miembro de la categoría @json. Todos los comandos JSON aplican cualquier restricción y permiso de espacio de teclas o comandos.
- Existen cinco categorías de ACL de Valkey o Redis OSS que se han actualizado para incluir los nuevos comandos de JSON: @read, @write, @fast, @slow y @admin. La siguiente tabla indica la asignación de los comandos JSON a las categorías apropiadas.

ACL

Comando JSON	@read	@write	@fast	@slow	@admin
JSON.ARRAPPEND		y	y		
JSON.ARRINDEX	y		y		
JSON.ARRINSERT		y	y		

Comando JSON	@read	@write	@fast	@slow	@admin
JSON.ARRLEN	y		y		
JSON.ARRPOP		y	y		
JSON.ARRTRIM		y	y		
JSON.CLEAR		y	y		
JSON.DEBUG	y			y	y
JSON.DEL		y	y		
JSON.FORGET		y	y		
JSON.GET	y		y		
JSON.MGET	y		y		
JSON.NUMINCRBY		y	y		
JSON.NUMMULTBY		y	y		
JSON.OBJKEYS	y		y		
JSON.OBJLEN	y		y		
JSON.RESP	y		y		

Comando JSON	@read	@write	@fast	@slow	@admin
JSON.SET		y		y	
JSON.STRAPPEND		y	y		
JSON.STRLEN	y		y		
JSON.STRLEN	y		y		
JSON.TOGGLE		y	y		
JSON.TYPE	y		y		
JSON.NUMINCRBY		y	y		

Límite de profundidad de anidado

Cuando un objeto o matriz JSON tiene un elemento que es otro objeto o matriz JSON, se dice que ese objeto o matriz interior se “anida” dentro del objeto o matriz exterior. El límite máximo de profundidad de anidamiento es 128. Cualquier intento de crear un documento que contenga una profundidad de anidamiento superior a 128 se rechazará con un error.

Sintaxis de comandos

La mayoría de los comandos requieren un nombre de clave como primer argumento. Algunos comandos también tienen un argumento ruta. El argumento ruta se establece por defecto en la raíz si es opcional y no proporcionado.

Notación:

- Los argumentos obligatorios se incluyen entre corchetes angulares. Por ejemplo: <key>
- Los argumentos opcionales deben ir entre corchetes. Por ejemplo: [path]

- Los argumentos opcionales adicionales se indican mediante puntos suspensivos ("..."). Por ejemplo: [json ...]

Sintaxis de ruta

Redis JSON admite dos tipos de sintaxis de rutas:

- Sintaxis mejorada: sigue la sintaxis JSONPath descrita por Goessner, como se muestra en la siguiente tabla. Hemos reordenado y modificado las descripciones de la tabla para mayor claridad.
- Sintaxis restringida: tiene capacidades de consulta limitadas.

> **Note**
>
> Los resultados de algunos comandos son sensibles al tipo de sintaxis de ruta que se utiliza.

Si una ruta de consulta comienza por '$', utiliza la sintaxis mejorada. De lo contrario, se utiliza la sintaxis restringida.

Sintaxis mejorada

Símbolo o expresión	Descripción
$	El elemento raíz.
. o bien []	Operador secundario.
..	Descenso recursivo.
*	Comodín. Todos los elementos de un objeto o matriz.
[]	Operador de subíndice de matriz. El índice se basa en 0.
[,]	Operador de unión.
[start:end:step]	Operador de Slice de la matriz.

Símbolo o expresión	Descripción
?()	Aplica una expresión de filtro (script) a la matriz u objeto actual.
()	Expresión de filtro.
@	Se usa en expresiones de filtro que hacen referencia al nodo actual que se está procesando.
==	Igual a, se utiliza en las expresiones de filtro.
!=	No es igual a, se utiliza en las expresiones de filtro.
>	Mayor que, se utiliza en las expresiones de filtro.
>=	Mayor o igual que, se utiliza en las expresiones de filtro.
<	Menor que, se utiliza en expresiones de filtro.
<=	Menor o igual que, se utiliza en las expresiones de filtro.
&&	Lógico Y, que se utiliza para combinar múltiples expresiones de filtro.
||	Lógico O, se utiliza para combinar múltiples expresiones de filtro.

Ejemplos

Los siguientes ejemplos se basan en los datos XML del ejemplo de Goessner, que hemos modificado agregando matrices adicionales.

```
{ "store": {
    "book": [
```

```
      { "category": "reference",
        "author": "Nigel Rees",
        "title": "Sayings of the Century",
        "price": 8.95,
        "in-stock": true,
        "sold": true
      },
      { "category": "fiction",
        "author": "Evelyn Waugh",
        "title": "Sword of Honour",
        "price": 12.99,
        "in-stock": false,
        "sold": true
      },
      { "category": "fiction",
        "author": "Herman Melville",
        "title": "Moby Dick",
        "isbn": "0-553-21311-3",
        "price": 8.99,
        "in-stock": true,
        "sold": false
      },
      { "category": "fiction",
        "author": "J. R. R. Tolkien",
        "title": "The Lord of the Rings",
        "isbn": "0-395-19395-8",
        "price": 22.99,
        "in-stock": false,
        "sold": false
      }
    ],
    "bicycle": {
      "color": "red",
      "price": 19.95,
      "in-stock": true,
      "sold": false
    }
  }
}
```

Ruta	Descripción
$.store.book[*].author	Los autores de todos los libros de la tienda.
$..author	Todos los autores.
$.store.*	Todos los miembros de la tienda.
$.store.*	Todos los miembros de la tienda.
$.store..price	El precio de todo lo que hay en la tienda.
$..*	Todos los miembros recursivos de la estructura JSON.
$..book[*]	Todos los libros.
$..book[0]	El primer libro.
$..book[-1]	El último libro.
$..book[0:2]	Los dos primeros libros.
$..book[0,1]	Los dos primeros libros.
$..book[0:4]	Los libros del índice 0 al 3 (el índice final no está incluido).
$..book[0:4:2]	Los libros en el índice 0, 2.
$..book[?(@.isbn)]	Todos los libros con un número de ISBN.
$..book[?(@.price<10)]	Todos los libros que cuestan menos de 10 dólares.
'$..book[?(@.price < 10)]'	Todos los libros que cuestan menos de 10 dólares. (La ruta debe estar entre comillas si contiene espacios en blanco).
'$..book[?(@["price"] < 10)]'	Todos los libros que cuestan menos de 10 dólares.

Ruta	Descripción
'$..book[?(@.["price"] < 10)]'	Todos los libros que cuestan menos de 10 dólares.
$..book[?(@.price>=10&&@.price<=100)]	Todos los libros en el rango de precios de 10 a 100 dólares, incluidos.
'$..book[?(@.price>=10 && @.price<=100)]'	Todos los libros en el rango de precios de 10 a 100 dólares, incluidos. (La ruta debe estar entre comillas si contiene espacios en blanco).
$..book[?(@.sold==true||@.in-stock==false)]	Todos los libros vendidos o agotados.
'$..book[?(@.sold == true || @.in-stock == false)]'	Todos los libros vendidos o agotados. (La ruta debe estar entre comillas si contiene espacios en blanco).
'$.store.book[?(@.["category"] == "fiction")]'	Todos los libros de la categoría Ficción.
'$.store.book[?(@.["category"] != "fiction")]'	Todos los libros de las categorías que no sean ficción.

Ejemplos de expresiones de filtro adicionales:

```
127.0.0.1:6379> JSON.SET k1 . '{"books": [{"price":5,"sold":true,"in-
stock":true,"title":"foo"}, {"price":15,"sold":false,"title":"abc"}]}'
OK
127.0.0.1:6379> JSON.GET k1 $.books[?(@.price>1&&@.price<20&&@.in-stock)]
"[{\"price\":5,\"sold\":true,\"in-stock\":true,\"title\":\"foo\"}]"
127.0.0.1:6379> JSON.GET k1 '$.books[?(@.price>1 && @.price<20 && @.in-stock)]'
"[{\"price\":5,\"sold\":true,\"in-stock\":true,\"title\":\"foo\"}]"
127.0.0.1:6379> JSON.GET k1 '$.books[?((@.price>1 && @.price<20) && (@.sold==false))]'
"[{\"price\":15,\"sold\":false,\"title\":\"abc\"}]"
127.0.0.1:6379> JSON.GET k1 '$.books[?(@.title == "abc")]'
[{"price":15,"sold":false,"title":"abc"}]

127.0.0.1:6379> JSON.SET k2 . '[1,2,3,4,5]'
127.0.0.1:6379> JSON.GET k2 $.*.[?(@>2)]
"[3,4,5]"
```

```
127.0.0.1:6379> JSON.GET k2 '$.*.[?(@ > 2)]'
"[3,4,5]"

127.0.0.1:6379> JSON.SET k3 . '[true,false,true,false,null,1,2,3,4]'
OK
127.0.0.1:6379> JSON.GET k3 $.*.[?(@==true)]
"[true,true]"
127.0.0.1:6379> JSON.GET k3 '$.*.[?(@ == true)]'
"[true,true]"
127.0.0.1:6379> JSON.GET k3 $.*.[?(@>1)]
"[2,3,4]"
127.0.0.1:6379> JSON.GET k3 '$.*.[?(@ > 1)]'
"[2,3,4]"
```

Sintaxis restringida

Símbolo o expresión	Descripción
. o bien []	Operador secundario.
[]	Operador de subíndice de matriz. El índice se basa en 0.

Ejemplos

Ruta	Descripción
.store.book[0].author	El autor del primer libro.
.store.book[-1].author	El autor del último libro.
.address.city	Nombre de la ciudad.
["store"]["book"][0]["title"]	El título del primer libro.
["store"]["book"][-1]["title"]	El título del último libro.

> **Note**
>
> Todo el contenido de Goessner citado en esta documentación está sujeto a la Licencia de Creative Commons.

Prefijos comunes de errores

Cada mensaje de error tiene un prefijo. A continuación se muestra una lista de prefijos comunes de errores.

Prefix	Descripción
ERR	Un error general.
LIMIT	Un error que se produce cuando se excede el límite de tamaño. Por ejemplo, cuando se excede el límite de tamaño del documento o el límite de profundidad de anidación.
INEXISTENTE	Una clave o ruta no existe.
FUERA DE LOS LÍMITES	Un índice de matrices fuera de los límites.
SYNTAXERR	Error de sintaxis.
WRONGTYPE	Tipo de valor incorrecto.

Métricas relacionadas con JSON

Se proporcionan las siguientes métricas de información JSON:

Información	Descripción
json_total_memory_bytes	Memoria total asignada a objetos JSON.
json_num_documents	El número total de documentos en Valkey o Redis OSS.

Para consultar las métricas principales, ejecute el siguiente comando:

```
info json_core_metrics
```

Cómo interacciona ElastiCache para Valkey y Redis OSS con JSON

En la siguiente sección, se describe cómo interactúa ElastiCache para Valkey y Redis OSS con el tipo de datos JSON.

Jerarquía de los operadores

Al evaluar las expresiones condicionales para el filtrado, las &&s tienen prioridad y, a continuación, se evalúan las ||s, como es común en la mayoría de los idiomas. Las operaciones entre paréntesis se ejecutan primero.

Comportamiento del límite máximo de anidación

El límite de anidación máximo de la ruta en ElastiCache para Redis OSS es 128. Así que un valor como `$.a.b.c.d...` solo puede alcanzar 128 niveles.

Administración de valores numéricos

JSON no tiene tipos de datos separados para números enteros y de coma flotante. Todos se llaman números.

Representaciones numéricas:

Cuando se recibe un número JSON en la entrada, se convierte en una de las dos representaciones binarias internas: un número entero firmado de 64 bits o un punto flotante de doble precisión IEEE de 64 bit. No se retiene la cadena original ni nada de su formato. Por lo tanto, cuando se genera un número como parte de una respuesta JSON, se convierte de la representación binaria interna a una cadena imprimible que utiliza reglas de formato genérico. Estas reglas podrían dan como resultado que se genere una cadena diferente de la que se recibió.

Comandos aritméticos `NUMINCRBY` y `NUMMULTBY`:

- Si ambos números son números enteros y el resultado está fuera del rango de `int64`, automáticamente se convierte en un número IEEE de punto flotante de doble precisión de 64 bits.
- Si al menos uno de los números es un punto flotante, el resultado es un número IEEE de punto flotante de doble precisión de 64 bits.

- Si el resultado supera el rango de doble IEEE de 64 bits, el comando regresa un error `OVERFLOW`.

Para obtener una lista de los comandos disponibles, consulte el Comandos de Valkey y Redis OSS compatibles.

Filtrado de matrices directas

ElastiCache para Valkey y Redis OSS filtra los objetos de matriz directamente.

Para datos como `[0,1,2,3,4,5,6]` y una consulta de ruta como `$[?(@<4)]`, o datos como `{"my_key":[0,1,2,3,4,5,6]}` y una consulta de ruta como `$.my_key[?(@<4)]`, ElastiCache devolvería [1,2,3] en ambas circunstancias.

Comportamiento de indexación de matrices

ElastiCache para Valkey o Redis OSS permite índices de respuestas positivas y negativas para matrices. Para una matriz de longitud cinco, 0 consultaría el primer elemento, 1 el segundo y, así, sucesivamente. Los números negativos comienzan al final de la matriz, por lo que -1 consultaría el quinto elemento, -2 el cuarto elemento, y así sucesivamente.

Para garantizar un comportamiento predecible para los clientes, ElastiCache no redondea los índices de matriz hacia abajo o hacia arriba, por lo que si tiene una matriz con una longitud de 5, que llama al índice 5 o superior, o -6 o inferior, no se generaría ningún resultado.

Evaluación de sintaxis estricta

MemoryDB no permite rutas JSON con sintaxis no válida, incluso si un subconjunto de la ruta contiene una ruta válida. Esto es para mantener un comportamiento correcto para nuestros clientes.

Comandos de Valkey y Redis OSS compatibles

ElastiCache admite los siguientes comandos de JSON en Valkey y Redis OSS:

Temas

- JSON.ARRAPPEND
- JSON.ARRINDEX
- JSON.ARRINSERT
- JSON.ARRLEN

- JSON.ARRPOP
- JSON.ARRTRIM
- JSON.CLEAR
- JSON.DEBUG
- JSON.DEL
- JSON.FORGET
- JSON.GET
- JSON.MGET
- JSON.MSET
- JSON.NUMINCRBY
- JSON.NUMMULTBY
- JSON.OBJLEN
- JSON.OBJKEYS
- JSON.RESP
- JSON.SET
- JSON.STRAPPEND
- JSON.STRLEN
- JSON.TOGGLE
- JSON.TYPE

JSON.ARRAPPEND

Adjunta uno o más valores a los valores de la matriz en la ruta.

Sintaxis

```
JSON.ARRAPPEND <key> <path> <json> [json ...]
```

- key (obligatorio): una clave de Valkey o Redis OSS del tipo de documento JSON.
- ruta (obligatorio): una ruta JSON.
- json (obligatorio): el valor JSON que se agregará a la matriz.

Return

Si la ruta es de sintaxis mejorada:

- Matriz de números enteros que representan la nueva longitud de la matriz en cada ruta.
- Si un valor no es una matriz, su valor devuelto correspondiente es nulo.
- `NONEXISTENT`Error si la ruta no existe.

Si la ruta es de sintaxis restringida:

- Entero, la nueva longitud de la matriz.
- Si se seleccionan varios valores de matriz, el comando devuelve la nueva longitud de la primera matriz actualizada.
- Error `WRONGTYPE` si el valor de la ruta no es una matriz.
- Error `SYNTAXERR` si uno de los argumentos json de entrada no es una cadena JSON válida.
- Error `NONEXISTENT` si la ruta no existe.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"]]'
OK
127.0.0.1:6379> JSON.ARRAPPEND  k1 $[*] '"c"'
1) (integer) 1
2) (integer) 2
3) (integer) 3
127.0.0.1:6379> JSON.GET k1
"[[\"c\"],[\"a\",\"c\"],[\"a\",\"b\",\"c\"]]"
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"]]'
OK
127.0.0.1:6379> JSON.ARRAPPEND  k1 [-1] '"c"'
(integer) 3
127.0.0.1:6379> JSON.GET k1
"[[],[\"a\"],[\"a\",\"b\",\"c\"]]"
```

JSON.ARRINDEX

Busca la primera aparición de un valor JSON escalar en las matrices de la ruta.

- Los errores fuera de rango se tratan redondeando el índice al principio y al final de la matriz.
- Si inicio > fin, devuelve -1 (no encontrado).

Sintaxis

```
JSON.ARRINDEX <key> <path> <json-scalar> [start [end]]
```

- key (obligatorio): una clave de Valkey o Redis OSS del tipo de documento JSON.
- ruta (obligatorio): una ruta JSON.
- json-scalar (obligatorio): el valor escalar para buscar . El escalar JSON se refiere a valores que no son objetos ni matrices. Es decir, cadena, número, Booleano y nulo son valores escalares.
- inicio (opcional): el índice de inicio, inclusivo. Toma 0 como valor predeterminado si no se proporciona.
- final (opcional): el índice final, exclusivo. Toma 0 como valor predeterminado si no se proporciona, lo que significa que se incluye el último elemento. 0 o -1 significa que se incluye el último elemento.

Return

Si la ruta es de sintaxis mejorada:

- Matriz de números enteros. Cada valor es el índice del elemento coincidente de la matriz en la ruta. El valor es -1 si no se encuentra.
- Si un valor no es una matriz, su valor devuelto correspondiente es nulo.

Si la ruta es de sintaxis restringida:

- Entero, el índice del elemento coincidente o -1 si no se encuentra.
- Error `WRONGTYPE` si el valor de la ruta no es una matriz.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"], ["a", "b", "c"]]'
OK
127.0.0.1:6379> JSON.ARRINDEX k1 $[*] '"b"'
1) (integer) -1
2) (integer) -1
3) (integer) 1
4) (integer) 1
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 . '{"children": ["John", "Jack", "Tom", "Bob", "Mike"]}'
OK
127.0.0.1:6379> JSON.ARRINDEX k1 .children '"Tom"'
(integer) 2
```

JSON.ARRINSERT

Inserta uno o varios valores en los valores de la matriz en la ruta antes del índice.

Sintaxis

```
JSON.ARRINSERT <key> <path> <index> <json> [json ...]
```

- key (obligatorio): una clave de Valkey o Redis OSS del tipo de documento JSON.
- ruta (obligatorio): una ruta JSON.
- índice (obligatorio): un índice de matriz antes del cual se insertan los valores.
- json (obligatorio): el valor JSON que se agregará a la matriz.

Return

Si la ruta es de sintaxis mejorada:

- Matriz de números enteros que representan la nueva longitud de la matriz en cada ruta.
- Si un valor es una matriz vacía, su valor devuelto correspondiente es nulo.
- Si un valor no es una matriz, su valor devuelto correspondiente es nulo.

- Error `OUTOFBOUNDARIES` si el argumento índice está fuera de los límites.

Si la ruta es de sintaxis restringida:

- Entero, la nueva longitud de la matriz.
- Error `WRONGTYPE` si el valor de la ruta no es una matriz.
- Error `OUTOFBOUNDARIES` si el argumento índice está fuera de los límites.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"]]'
OK
127.0.0.1:6379> JSON.ARRINSERT k1 $[*] 0 '"c"'
1) (integer) 1
2) (integer) 2
3) (integer) 3
127.0.0.1:6379> JSON.GET k1
"[[\"c\"],[\"c\",\"a\"],[\"c\",\"a\",\"b\"]]"
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"]]'
OK
127.0.0.1:6379> JSON.ARRINSERT k1 . 0 '"c"'
(integer) 4
127.0.0.1:6379> JSON.GET k1
"[\"c\",[],[\"a\"],[\"a\",\"b\"]]"
```

JSON.ARRLEN

Consigue la longitud de los valores de la matriz en la ruta.

Sintaxis

```
JSON.ARRLEN <key> [path]
```

- key (obligatorio): una clave de Valkey o Redis OSS del tipo de documento JSON.
- ruta (opcional): una ruta JSON. Toma el valor predeterminado raíz si no se proporciona.

Return

Si la ruta es de sintaxis mejorada:

- Matriz de números enteros que representa la longitud de la matriz en cada ruta.
- Si un valor no es una matriz, su valor devuelto correspondiente es nulo.
- Es nulo si la clave del documento no existe.

Si la ruta es de sintaxis restringida:

- Entero, longitud de matriz.
- Si hay varios objetos seleccionados, el comando devuelve la longitud de la primera matriz.
- Error `WRONGTYPE` si el valor de la ruta no es una matriz.
- `NONEXISTENT JSON`Error si la ruta no existe.
- Es nulo si la clave del documento no existe.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"], ["a", "b", "c"]]'
OK
127.0.0.1:6379> JSON.ARRLEN k1 $[*]
1) (integer) 0
2) (integer) 1
3) (integer) 2
4) (integer) 3

127.0.0.1:6379> JSON.SET k2 . '[[], "a", ["a", "b"], ["a", "b", "c"], 4]'
OK
127.0.0.1:6379> JSON.ARRLEN k2 $[*]
1) (integer) 0
2) (nil)
3) (integer) 2
4) (integer) 3
```

```
5) (nil)
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"], ["a", "b", "c"]]'
OK
127.0.0.1:6379> JSON.ARRLEN k1 [*]
(integer) 0
127.0.0.1:6379> JSON.ARRLEN k1 [1]
(integer) 1
127.0.0.1:6379> JSON.ARRLEN k1 [2]
(integer) 2

127.0.0.1:6379> JSON.SET k2 . '[[], "a", ["a", "b"], ["a", "b", "c"], 4]'
OK
127.0.0.1:6379> JSON.ARRLEN k2 [1]
(error) WRONGTYPE JSON element is not an array
127.0.0.1:6379> JSON.ARRLEN k2 [0]
(integer) 0
127.0.0.1:6379> JSON.ARRLEN k2 [6]
(error) OUTOFBOUNDARIES Array index is out of bounds
127.0.0.1:6379> JSON.ARRLEN k2 a.b
(error) NONEXISTENT JSON path does not exist
```

JSON.ARRPOP

Elimina y devuelve elemento en el índice de la matriz. Al emerger una matriz vacía, se devuelve nulo.

Sintaxis

```
JSON.ARRPOP <key> [path [index]]
```

- key (obligatorio): una clave de Valkey o Redis OSS del tipo de documento JSON.
- ruta (opcional): una ruta JSON. Toma el valor predeterminado raíz si no se proporciona.
- índice (opcional): la pposición en la matriz desde la que empezar a salir.
  - El valor predeterminado es -1 si no se proporciona, lo que significa el último elemento.
  - Un valor negativo significa la posición desde el último elemento.
  - Los índices fuera de los límites se redondean a sus respectivos límites de matriz.

Return

Si la ruta es de sintaxis mejorada:

- Matriz de cadenas masivas que representan valores emergentes en cada ruta.
- Si un valor es una matriz vacía, su valor devuelto correspondiente es nulo.
- Si un valor no es una matriz, su valor devuelto correspondiente es nulo.

Si la ruta es de sintaxis restringida:

- Cadena por lotes, que representa el valor JSON emergente.
- Es nulo si la matriz está vacía.
- Error `WRONGTYPE` si el valor de la ruta no es una matriz.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"]]'
OK
127.0.0.1:6379> JSON.ARRPOP k1 $[*]
1) (nil)
2) "\"a\""
3) "\"b\""
127.0.0.1:6379> JSON.GET k1
"[[],[],[\"a\"]]"
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"]]'
OK
127.0.0.1:6379> JSON.ARRPOP k1
"[\"a\",\"b\"]"
127.0.0.1:6379> JSON.GET k1
"[[],[\"a\"]]"

127.0.0.1:6379> JSON.SET k2 . '[[], ["a"], ["a", "b"]]'
OK
```

```
127.0.0.1:6379> JSON.ARRPOP k2 . 0
"[]"
127.0.0.1:6379> JSON.GET k2
"[[\"a\"],[\"a\",\"b\"]]"
```

JSON.ARRTRIM

Recorta una matriz en la ruta para que se convierta en un subcampo [inicio, fin], ambos inclusivos.

- Si la matriz está vacía, no se hace nada y se devuelve 0.
- Si el valor inicio es <0, trátelo como 0.
- Si el tamaño del valor final es >= (tamaño de la matriz), trátelo como tamaño-1.
- Si el tamaño del valor inicio >= o inicio > final, vacíe la matriz y devuelva 0.

Sintaxis

```
JSON.ARRTRIM <key> <path> <start> <end>
```

- key (obligatorio): una clave de Valkey o Redis OSS del tipo de documento JSON.
- ruta (obligatorio): una ruta JSON.
- inicio (obligatorio): el índice de inicio, inclusivo.
- final (obligatorio): el índice de final, inclusivo.

Return

Si la ruta es de sintaxis mejorada:

- Matriz de números enteros que representan la nueva longitud de la matriz en cada ruta.
- Si un valor es una matriz vacía, su valor devuelto correspondiente es nulo.
- Si un valor no es una matriz, su valor devuelto correspondiente es nulo.
- Error `OUTOFBOUNDARIES` si un argumento de índice está fuera de los límites.

Si la ruta es de sintaxis restringida:

- Entero, la nueva longitud de la matriz.

- Es nulo si la matriz está vacía.
- Error `WRONGTYPE` si el valor de la ruta no es una matriz.
- Error `OUTOFBOUNDARIES` si un argumento de índice está fuera de los límites.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"], ["a", "b", "c"]]'
OK
127.0.0.1:6379> JSON.ARRTRIM k1 $[*] 0 1
1) (integer) 0
2) (integer) 1
3) (integer) 2
4) (integer) 2
127.0.0.1:6379> JSON.GET k1
"[[],[\"a\"],[\"a\",\"b\"],[\"a\",\"b\"]]"
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 . '{"children": ["John", "Jack", "Tom", "Bob", "Mike"]}'
OK
127.0.0.1:6379> JSON.ARRTRIM k1 .children 0 1
(integer) 2
127.0.0.1:6379> JSON.GET k1 .children
"[\"John\",\"Jack\"]"
```

JSON.CLEAR

Borra las matrices o un objeto en la ruta.

Sintaxis

```
JSON.CLEAR <key> [path]
```

- key (obligatorio): una clave de Valkey o Redis OSS del tipo de documento JSON.
- ruta (opcional): una ruta JSON. Toma el valor predeterminado raíz si no se proporciona.

Return

- Entero, el número de contenedores borrados.
- La eliminación de una matriz u objeto vacío representa 1 contenedor borrado.
- Al borrar un valor no contenedor, se devuelve 0.

Ejemplos

```
127.0.0.1:6379> JSON.SET k1 . '[[], [0], [0,1], [0,1,2], 1, true, null, "d"]'
OK
127.0.0.1:6379>  JSON.CLEAR k1  $[*]
(integer) 7
127.0.0.1:6379> JSON.CLEAR k1  $[*]
(integer) 4
127.0.0.1:6379> JSON.SET k2 . '{"children": ["John", "Jack", "Tom", "Bob", "Mike"]}'
OK
127.0.0.1:6379> JSON.CLEAR k2 .children
(integer) 1
127.0.0.1:6379> JSON.GET k2 .children
"[]"
```

JSON.DEBUG

Reporta información. Los subcomandos admitidos son:

- MEMORIA <key> [path]: Informa el uso de memoria en bytes de un valor JSON. La ruta se establece de forma predeterminada en la raíz si no se proporciona.
- CAMPOS <key> [path]: Informa el número de campos de la ruta del documento especificada. La ruta se establece de forma predeterminada en la raíz si no se proporciona. Cada valor JSON que no es de contenedor cuenta como un campo. Los objetos y las matrices cuentan de forma recursiva un campo para cada uno de los valores JSON que contienen. Cada valor de contenedor, excepto el contenedor raíz, cuenta como un campo adicional.
- AYUDA: imprime mensajes de ayuda del comando.

Sintaxis

```
JSON.DEBUG <subcommand & arguments>
```

Depende del subcomando:

MEMORIA

- Si la ruta es de sintaxis mejorada:
  - Devuelve una matriz de números enteros que representan tamaño de memoria (en bytes) del valor de JSON en cada ruta.
  - Devuelve una matriz vacía si la clave de Valkey o Redis OSS no existe.
- Si la ruta es de sintaxis restringida:
  - Devuelve un número entero, tamaño de memoria y el valor JSON en bytes.
  - Devuelve null si la clave de Valkey o Redis OSS no existe.

FIELDS

- Si la ruta es de sintaxis mejorada:
  - Devuelve una matriz de números enteros que representar el número de campos de valor JSON en cada ruta.
  - Devuelve una matriz vacía si la clave de Valkey o Redis OSS no existe.
- Si la ruta es de sintaxis restringida:
  - Devuelve un número entero, el número de campos del valor JSON.
  - Devuelve null si la clave de Valkey o Redis OSS no existe.

AYUDA: devuelve una serie de mensajes de ayuda.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 . '[1, 2.3, "foo", true, null, {}, [], {"a":1, "b":2},
 [1,2,3]]'
OK
127.0.0.1:6379> JSON.DEBUG MEMORY k1 $[*]
1) (integer) 16
2) (integer) 16
3) (integer) 19
4) (integer) 16
5) (integer) 16
```

```
6) (integer) 16
7) (integer) 16
8) (integer) 50
9) (integer) 64
127.0.0.1:6379> JSON.DEBUG FIELDS k1 $[*]
1) (integer) 1
2) (integer) 1
3) (integer) 1
4) (integer) 1
5) (integer) 1
6) (integer) 0
7) (integer) 0
8) (integer) 2
9) (integer) 3
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 .
 '{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
{"street":"21 2nd Street","city":"New
 York","state":"NY","zipcode":"10021-3100"},"phoneNumbers":
[{"type":"home","number":"212 555-1234"},{"type":"office","number":"646
 555-4567"}],"children":[],"spouse":null}'
OK
127.0.0.1:6379> JSON.DEBUG MEMORY k1
(integer) 632
127.0.0.1:6379> JSON.DEBUG MEMORY k1 .phoneNumbers
(integer) 166

127.0.0.1:6379> JSON.DEBUG FIELDS k1
(integer) 19
127.0.0.1:6379> JSON.DEBUG FIELDS k1 .address
(integer) 4

127.0.0.1:6379> JSON.DEBUG HELP
1) JSON.DEBUG MEMORY <key> [path] - report memory size (bytes) of the JSON element.
 Path defaults to root if not provided.
2) JSON.DEBUG FIELDS <key> [path] - report number of fields in the JSON element. Path
 defaults to root if not provided.
3) JSON.DEBUG HELP - print help message.
```

JSON.DEL

Borra los valores JSON de la ruta de acceso de una clave de documento. Si la ruta es la raíz, equivale a eliminar la clave de Valkey o Redis OSS.

Sintaxis

```
JSON.DEL <key> [path]
```

- key (obligatorio): una clave de Valkey o Redis OSS del tipo de documento JSON.
- ruta (opcional): una ruta JSON. Toma el valor predeterminado raíz si no se proporciona.

Return

- Número de elementos eliminados.
- Es 0 si la clave de Valkey o Redis OSS no existe.
- 0 si la ruta JSON no es válida o no existe.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1,
 "b":2, "c":3}, "e": [1,2,3,4,5]}'
OK
127.0.0.1:6379> JSON.DEL k1 $.d.*
(integer) 3
127.0.0.1:6379> JSOn.GET k1
"{\"a\":{},\"b\":{\"a\":1},\"c\":{\"a\":1,\"b\":2},\"d\":{},\"e\":[1,2,3,4,5]}"
127.0.0.1:6379> JSON.DEL k1 $.e[*]
(integer) 5
127.0.0.1:6379> JSOn.GET k1
"{\"a\":{},\"b\":{\"a\":1},\"c\":{\"a\":1,\"b\":2},\"d\":{},\"e\":[]}"
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1,
 "b":2, "c":3}, "e": [1,2,3,4,5]}'
```

```
OK
127.0.0.1:6379> JSON.DEL k1 .d.*
(integer) 3
127.0.0.1:6379> JSON.GET k1
"{\"a\":{},\"b\":{\"a\":1},\"c\":{\"a\":1,\"b\":2},\"d\":{},\"e\":[1,2,3,4,5]}"
127.0.0.1:6379> JSON.DEL k1 .e[*]
(integer) 5
127.0.0.1:6379> JSON.GET k1
"{\"a\":{},\"b\":{\"a\":1},\"c\":{\"a\":1,\"b\":2},\"d\":{},\"e\":[]}"
```

JSON.FORGET

Un alias de JSON.DEL.

JSON.GET

Devuelve el formato JSON serializado en una o varias rutas.

Sintaxis

```
JSON.GET <key>
[INDENT indentation-string]
[NEWLINE newline-string]
[SPACE space-string]
[NOESCAPE]
[path ...]
```

- key (obligatorio): una clave de Valkey o Redis OSS del tipo de documento JSON.
- SANGRÍA/NUEVA LÍNEA/ESPACIO (opcional): controla el formato de la cadena de JSON devuelta, es decir, “impresión bonita”. El valor predeterminado de cada una es una cadena vacía. Se puede anular en cualquier combinación. Estos se pueden especificar en cualquier orden.
- SIN ESCAPE: opcional, puede estar presente para la compatibilidad con versiones anteriores y no tiene ningún otro efecto.
- ruta (opcional): cero o más rutas JSON, el valor predeterminado es la raíz si no se proporciona ninguna. Los argumentos de la ruta deben colocarse al final.

Return

Sintaxis de la ruta mejorada:

Si se da una ruta:

- Devuelve una cadena serializada de una matriz de valores.
- Si no selecciona ningún valor, el comando devuelve una matriz vacía.

Si se proporcionan varias rutas:

- Devuelve un objeto JSON con cadenas, en el que cada ruta es una clave.
- Si hay una sintaxis de ruta restringida y mejorada mixta, el resultado se ajusta a la sintaxis mejorada.
- Si no existe una ruta, su valor correspondiente es una matriz vacía.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 .
 '{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
{"street":"21 2nd Street","city":"New
 York","state":"NY","zipcode":"10021-3100"},"phoneNumbers":
[{"type":"home","number":"212 555-1234"},{"type":"office","number":"646
 555-4567"}],"children":[],"spouse":null}'
OK
127.0.0.1:6379> JSON.GET k1 $.address.*
"[\"21 2nd Street\",\"New York\",\"NY\",\"10021-3100\"]"
127.0.0.1:6379> JSON.GET k1 indent "\t" space " " NEWLINE "\n" $.address.*
"[\n\t\"21 2nd Street\",\n\t\"New York\",\n\t\"NY\",\n\t\"10021-3100\"\n]"
127.0.0.1:6379> JSON.GET k1 $.firstName $.lastName $.age
"{\"$.firstName\":[\"John\"],\"$.lastName\":[\"Smith\"],\"$.age\":[27]}"
127.0.0.1:6379> JSON.SET k2 . '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}}'
OK
127.0.0.1:6379> json.get k2 $..*
"[{},{\"a\":1},{\"a\":1,\"b\":2},1,1,2]"
```

Sintaxis de la ruta restringida:

```
 127.0.0.1:6379> JSON.SET k1 .
 '{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
```

```
{"street":"21 2nd Street","city":"New
 York","state":"NY","zipcode":"10021-3100"},"phoneNumbers":
[{"type":"home","number":"212 555-1234"},{"type":"office","number":"646
 555-4567"}],"children":[],"spouse":null}'
OK
127.0.0.1:6379> JSON.GET k1 .address
"{\"street\":\"21 2nd Street\",\"city\":\"New York\",\"state\":\"NY\",\"zipcode\":
\"10021-3100\"}"
127.0.0.1:6379> JSON.GET k1 indent "\t" space " " NEWLINE "\n" .address
"{\n\t\"street\": \"21 2nd Street\",\n\t\"city\": \"New York\",\n\t\"state\": \"NY\",\n
\t\"zipcode\": \"10021-3100\"\n}"
127.0.0.1:6379> JSON.GET k1 .firstName .lastName .age
"{\".firstName\":\"John\",\".lastName\":\"Smith\",\".age\":27}"
```

JSON.MGET

Consigue los comandos JSON serializados en la ruta de varias claves de documentos. Devuelve un valor nulo para una clave o ruta JSON no existente.

Sintaxis

```
JSON.MGET <key> [key ...] <path>
```

- key (obligatorio): una o más claves de Valkey o Redis OSS del tipo de documento.
- ruta (obligatorio): una ruta JSON.

Return

- Matriz de cadenas a granel. El tamaño de la matriz es igual al número de teclas del comando. Cada elemento de la matriz se rellena con (a) el comando JSON serializado tal como se encuentra en la ruta o (b) nulo si la clave no existe, la ruta no existe en el documento, o la ruta no es válida (error de sintaxis).
- Si alguna de las claves especificadas existe y no es una clave JSON, el comando devuelve el error `WRONGTYPE`.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 . '{"address":{"street":"21 2nd Street","city":"New
 York","state":"NY","zipcode":"10021"}}'
OK
127.0.0.1:6379> JSON.SET k2 . '{"address":{"street":"5 main
 Street","city":"Boston","state":"MA","zipcode":"02101"}}'
OK
127.0.0.1:6379> JSON.SET k3 . '{"address":{"street":"100 Park
 Ave","city":"Seattle","state":"WA","zipcode":"98102"}}'
OK
127.0.0.1:6379> JSON.MGET k1 k2 k3 $.address.city
1) "[\"New York\"]"
2) "[\"Boston\"]"
3) "[\"Seattle\"]"
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 . '{"address":{"street":"21 2nd Street","city":"New
 York","state":"NY","zipcode":"10021"}}'
OK
127.0.0.1:6379> JSON.SET k2 . '{"address":{"street":"5 main
 Street","city":"Boston","state":"MA","zipcode":"02101"}}'
OK
127.0.0.1:6379> JSON.SET k3 . '{"address":{"street":"100 Park
 Ave","city":"Seattle","state":"WA","zipcode":"98102"}}'
OK

127.0.0.1:6379> JSON.MGET k1 k2 k3 .address.city
1) "\"New York\""
2) "\"Seattle\""
3) "\"Seattle\""
```

JSON.MSET

Compatible con la versión 8.1 y posteriores de Valkey.

Establezca los valores JSON para varias claves. La operación es atómica. O bien se establecen todos los valores o ninguno.

Sintaxis

```
JSON.MSET key path json [ key path json ... ]
```

- Si la ruta de acceso llama a un miembro de objeto:
  - Si el elemento principal no existe, el comando devolverá un error INEXISTENTE.
  - Si el elemento principal existe pero no es un objeto, el comando devolverá ERROR.
  - Si el elemento principal existe y es un objeto:
    - Si el miembro no existe, se anexará un miembro nuevo al objeto principal si y solo si el objeto principal es el último objeto secundario de la ruta. De lo contrario, el comando devolverá un error INEXISTENTE.
    - Si el miembro existe, su valor se reemplazará por el valor JSON.
- Si la ruta requiere un índice de matriz:
  - Si el elemento principal no existe, el comando devolverá un error INEXISTENTE.
  - Si el elemento principal existe pero no es una matriz, el comando devolverá ERROR.
  - Si el elemento principal existe pero el índice está fuera de los límites, el comando devuelve un error OUTOFBOUNDARIES.
  - Si el elemento principal existe y el índice es válido, el elemento se reemplazará por el nuevo valor JSON.
- Si la ruta llama a un objeto o matriz, el valor (objeto o matriz) se reemplazará por el nuevo valor JSON.

Return

- Respuesta de cadena simple: OK si la operación se ha realizado correctamente.
- Respuesta de error simple: si la operación no se ha ejecutado correctamente.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.MSET k1 . '[1,2,3,4,5]' k2 . '{"a":{"a":1, "b":2, "c":3}}' k3 .
 '{"a": [1,2,3,4,5]}'
OK
127.0.0.1:6379> JSON.GET k1
"[1,2,3,4,5]"
127.0.0.1:6379> JSON.GET k2
```

```
"{\"a\":{\"a\":1,\"b\":2,\"c\":3}}"
127.0.0.1:6379> JSON.MSET k2 $.a.* '0' k3 $.a[*] '0'
OK
127.0.0.1:6379> JSON.GET k2
"{\"a\":{\"a\":0,\"b\":0,\"c\":0}}"
127.0.0.1:6379> JSON.GET k3
"{\"a\":[0,0,0,0,0]}"
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.MSET k1 . '{"name": "John","address": {"street": "123 Main
 St","city": "Springfield"},"phones": ["555-1234","555-5678"]}'
OK
127.0.0.1:6379> JSON.MSET k1 .address.street '"21 2nd Street"' k1 .address.city '"New
 York"'
OK
127.0.0.1:6379> JSON.GET k1 .address.street
"\"21 2nd Street\""
127.0.0.1:6379> JSON.GET k1 .address.city
"\"New York\""
```

JSON.NUMINCRBY

Aumenta los valores numéricos de la ruta por un número dado.

Sintaxis

```
JSON.NUMINCRBY <key> <path> <number>
```

- key (obligatorio): una clave de Valkey o Redis OSS del tipo de documento JSON.
- ruta (obligatorio): una ruta JSON.
- número (obligatorio): un número.

Return

Si la ruta es de sintaxis mejorada:

- Matriz de cadena a granel que representa el valor resultante en cada ruta.
- Si un valor no es un número, su valor devuelto correspondiente es nulo.
- El error `WRONGTYPE` si el número no se puede analizar.

- El error `OVERFLOW` si el resultado está fuera del rango del doble IEEE de 64 bits.
- `NONEXISTENT` si la clave del documento no existe.

Si la ruta es de sintaxis restringida:

- Cadena a granel que representa el valor resultante.
- Si se seleccionan varios valores, el comando devuelve el resultado del último valor actualizado.
- El error `WRONGTYPE` si el valor de la ruta no es un número.
- El error `WRONGTYPE` si el número no se puede analizar.
- El error `OVERFLOW` si el resultado está fuera del rango del doble IEEE de 64 bits.
- `NONEXISTENT` si la clave del documento no existe.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k1 $.d[*] 10
"[11,12,13]"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[1],\"c\":[1,2],\"d\":[11,12,13]}"

127.0.0.1:6379> JSON.SET k1 $ '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k1 $.a[*] 1
"[]"
127.0.0.1:6379> JSON.NUMINCRBY k1 $.b[*] 1
"[2]"
127.0.0.1:6379> JSON.NUMINCRBY k1 $.c[*] 1
"[2,3]"
127.0.0.1:6379> JSON.NUMINCRBY k1 $.d[*] 1
"[2,3,4]"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[2,3],\"d\":[2,3,4]}"

127.0.0.1:6379> JSON.SET k2 $ '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1,
 "b":2, "c":3}}'
OK
```

```
127.0.0.1:6379> JSON.NUMINCRBY k2 $.a.* 1
"[]"
127.0.0.1:6379> JSON.NUMINCRBY k2 $.b.* 1
"[2]"
127.0.0.1:6379> JSON.NUMINCRBY k2 $.c.* 1
"[2,3]"
127.0.0.1:6379> JSON.NUMINCRBY k2 $.d.* 1
"[2,3,4]"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{},\"b\":{\"a\":2},\"c\":{\"a\":2,\"b\":3},\"d\":{\"a\":2,\"b\":3,\"c\":4}}"

127.0.0.1:6379> JSON.SET k3 $ '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
 "b":"b"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k3 $.a.* 1
"[null]"
127.0.0.1:6379> JSON.NUMINCRBY k3 $.b.* 1
"[null,2]"
127.0.0.1:6379> JSON.NUMINCRBY k3 $.c.* 1
"[null,null]"
127.0.0.1:6379> JSON.NUMINCRBY k3 $.d.* 1
"[2,null,4]"
127.0.0.1:6379> JSON.GET k3
"{\"a\":{\"a\":\"a\"},\"b\":{\"a\":\"a\",\"b\":2},\"c\":{\"a\":\"a\",\"b\":\"b\"},\"d
\":{\"a\":2,\"b\":\"b\",\"c\":4}}"
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k1 .d[1] 10
"12"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[1],\"c\":[1,2],\"d\":[1,12,3]}"

127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k1 .a[*] 1
(error) NONEXISTENT JSON path does not exist
127.0.0.1:6379> JSON.NUMINCRBY k1 .b[*] 1
"2"
127.0.0.1:6379> JSON.GET k1
```

```
"{\"a\":[],\"b\":[2],\"c\":[1,2],\"d\":[1,2,3]}"
127.0.0.1:6379> JSON.NUMINCRBY k1 .c[*] 1
"3"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[2,3],\"d\":[1,2,3]}"
127.0.0.1:6379> JSON.NUMINCRBY k1 .d[*] 1
"4"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[2,3],\"d\":[2,3,4]}"

127.0.0.1:6379> JSON.SET k2 . '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1,
 "b":2, "c":3}}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k2 .a.* 1
(error) NONEXISTENT JSON path does not exist
127.0.0.1:6379> JSON.NUMINCRBY k2 .b.* 1
"2"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{},\"b\":{\"a\":2},\"c\":{\"a\":1,\"b\":2},\"d\":{\"a\":1,\"b\":2,\"c\":3}}"
127.0.0.1:6379> JSON.NUMINCRBY k2 .c.* 1
"3"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{},\"b\":{\"a\":2},\"c\":{\"a\":2,\"b\":3},\"d\":{\"a\":1,\"b\":2,\"c\":3}}"
127.0.0.1:6379> JSON.NUMINCRBY k2 .d.* 1
"4"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{},\"b\":{\"a\":2},\"c\":{\"a\":2,\"b\":3},\"d\":{\"a\":2,\"b\":3,\"c\":4}}"

127.0.0.1:6379> JSON.SET k3 . '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
 "b":"b"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k3 .a.* 1
(error) WRONGTYPE JSON element is not a number
127.0.0.1:6379> JSON.NUMINCRBY k3 .b.* 1
"2"
127.0.0.1:6379> JSON.NUMINCRBY k3 .c.* 1
(error) WRONGTYPE JSON element is not a number
127.0.0.1:6379> JSON.NUMINCRBY k3 .d.* 1
"4"
```

JSON.NUMMULTBY

Multipla los valores numéricos de la ruta por un número dado.

Sintaxis

```
JSON.NUMMULTBY <key> <path> <number>
```

- key (obligatorio): una clave de Valkey o Redis OSS del tipo de documento JSON.
- ruta (obligatorio): una ruta JSON.
- número (obligatorio): un número.

Return

Si la ruta es de sintaxis mejorada:

- Matriz de cadenas a granel que representa el valor resultante en cada ruta.
- Si un valor no es un número, su valor devuelto correspondiente es nulo.
- El error `WRONGTYPE` si el número no se puede analizar.
- El error `OVERFLOW` si el resultado está fuera del rango de un número de punto flotante de doble precisión de 64 bits IEEE.
- `NONEXISTENT` si la clave del documento no existe.

Si la ruta es de sintaxis restringida:

- Cadena a granel que representa el valor resultante.
- Si se seleccionan varios valores, el comando devuelve el resultado del último valor actualizado.
- El error `WRONGTYPE` si el valor de la ruta no es un número.
- El error `WRONGTYPE` si el número no se puede analizar.
- El error `OVERFLOW` si el resultado está fuera del rango del doble IEEE de 64 bits.
- `NONEXISTENT` si la clave del documento no existe.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k1 $.d[*] 2
"[2,4,6]"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[1],\"c\":[1,2],\"d\":[2,4,6]}"

127.0.0.1:6379> JSON.SET k1 $ '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k1 $.a[*] 2
"[]"
127.0.0.1:6379> JSON.NUMMULTBY k1 $.b[*] 2
"[2]"
127.0.0.1:6379> JSON.NUMMULTBY k1 $.c[*] 2
"[2,4]"
127.0.0.1:6379> JSON.NUMMULTBY k1 $.d[*] 2
"[2,4,6]"

127.0.0.1:6379> JSON.SET k2 $ '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1,
 "b":2, "c":3}}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k2 $.a.* 2
"[]"
127.0.0.1:6379> JSON.NUMMULTBY k2 $.b.* 2
"[2]"
127.0.0.1:6379> JSON.NUMMULTBY k2 $.c.* 2
"[2,4]"
127.0.0.1:6379> JSON.NUMMULTBY k2 $.d.* 2
"[2,4,6]"

127.0.0.1:6379> JSON.SET k3 $ '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
 "b":"b"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k3 $.a.* 2
"[null]"
127.0.0.1:6379> JSON.NUMMULTBY k3 $.b.* 2
"[null,2]"
127.0.0.1:6379> JSON.NUMMULTBY k3 $.c.* 2
"[null,null]"
127.0.0.1:6379> JSON.NUMMULTBY k3 $.d.* 2
"[2,null,6]"
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k1 .d[1] 2
"4"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[1],\"c\":[1,2],\"d\":[1,4,3]}"

127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k1 .a[*] 2
(error) NONEXISTENT JSON path does not exist
127.0.0.1:6379> JSON.NUMMULTBY k1 .b[*] 2
"2"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[1,2],\"d\":[1,2,3]}"
127.0.0.1:6379> JSON.NUMMULTBY k1 .c[*] 2
"4"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[2,4],\"d\":[1,2,3]}"
127.0.0.1:6379> JSON.NUMMULTBY k1 .d[*] 2
"6"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[2,4],\"d\":[2,4,6]}"

127.0.0.1:6379> JSON.SET k2 . '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1,
 "b":2, "c":3}}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k2 .a.* 2
(error) NONEXISTENT JSON path does not exist
127.0.0.1:6379> JSON.NUMMULTBY k2 .b.* 2
"2"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{},\"b\":{\"a\":2},\"c\":{\"a\":1,\"b\":2},\"d\":{\"a\":1,\"b\":2,\"c\":3}}"
127.0.0.1:6379> JSON.NUMMULTBY k2 .c.* 2
"4"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{},\"b\":{\"a\":2},\"c\":{\"a\":2,\"b\":4},\"d\":{\"a\":1,\"b\":2,\"c\":3}}"
127.0.0.1:6379> JSON.NUMMULTBY k2 .d.* 2
"6"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{},\"b\":{\"a\":2},\"c\":{\"a\":2,\"b\":4},\"d\":{\"a\":2,\"b\":4,\"c\":6}}"
```

```
127.0.0.1:6379> JSON.SET k3 . '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
 "b":"b"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k3 .a.* 2
(error) WRONGTYPE JSON element is not a number
127.0.0.1:6379> JSON.NUMMULTBY k3 .b.* 2
"2"
127.0.0.1:6379> JSON.GET k3
"{\"a\":{\"a\":\"a\"},\"b\":{\"a\":\"a\",\"b\":2},\"c\":{\"a\":\"a\",\"b\":\"b\"},\"d
\":{\"a\":1,\"b\":\"b\",\"c\":3}}"
127.0.0.1:6379> JSON.NUMMULTBY k3 .c.* 2
(error) WRONGTYPE JSON element is not a number
127.0.0.1:6379> JSON.NUMMULTBY k3 .d.* 2
"6"
127.0.0.1:6379> JSON.GET k3
"{\"a\":{\"a\":\"a\"},\"b\":{\"a\":\"a\",\"b\":2},\"c\":{\"a\":\"a\",\"b\":\"b\"},\"d
\":{\"a\":2,\"b\":\"b\",\"c\":6}}"
```

JSON.OBJLEN

Consigue el número de claves en los valores del objeto en la ruta.

Sintaxis

```
JSON.OBJLEN <key> [path]
```

- key (obligatorio): una clave de Valkey o Redis OSS del tipo de documento JSON.
- ruta (opcional): una ruta JSON. Toma el valor predeterminado raíz si no se proporciona.

Return

Si la ruta es de sintaxis mejorada:

- Matriz de números enteros que representa la longitud del objeto en cada ruta.
- Si un valor no es un objeto, su valor devuelto correspondiente es nulo.
- Es nulo si la clave del documento no existe.

Si la ruta es de sintaxis restringida:

- Entero, número de claves del objeto.
- Si hay varios objetos seleccionados, el comando devuelve la longitud del primer objeto.
- El error `WRONGTYPE` si el valor de la ruta no es un objeto.
- `NONEXISTENT JSON`Error si la ruta no existe.
- Es nulo si la clave del documento no existe.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{}, "b":{"a":"a"}, "c":{"a":"a", "b":"bb"}, "d":
{"a":1, "b":"b", "c":{"a":3,"b":4}}, "e":1}'
OK
127.0.0.1:6379> JSON.OBJLEN k1 $.a
1) (integer) 0
127.0.0.1:6379> JSON.OBJLEN k1 $.a.*
(empty array)
127.0.0.1:6379> JSON.OBJLEN k1 $.b
1) (integer) 1
127.0.0.1:6379> JSON.OBJLEN k1 $.b.*
1) (nil)
127.0.0.1:6379> JSON.OBJLEN k1 $.c
1) (integer) 2
127.0.0.1:6379> JSON.OBJLEN k1 $.c.*
1) (nil)
2) (nil)
127.0.0.1:6379> JSON.OBJLEN k1 $.d
1) (integer) 3
127.0.0.1:6379> JSON.OBJLEN k1 $.d.*
1) (nil)
2) (nil)
3) (integer) 2
127.0.0.1:6379> JSON.OBJLEN k1 $.*
1) (integer) 0
2) (integer) 1
3) (integer) 2
4) (integer) 3
5) (nil)
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":{}, "b":{"a":"a"}, "c":{"a":"a", "b":"bb"}, "d":
{"a":1, "b":"b", "c":{"a":3,"b":4}}, "e":1}'
OK
127.0.0.1:6379> JSON.OBJLEN k1 .a
(integer) 0
127.0.0.1:6379> JSON.OBJLEN k1 .a.*
(error) NONEXISTENT JSON path does not exist
127.0.0.1:6379> JSON.OBJLEN k1 .b
(integer) 1
127.0.0.1:6379> JSON.OBJLEN k1 .b.*
(error) WRONGTYPE JSON element is not an object
127.0.0.1:6379> JSON.OBJLEN k1 .c
(integer) 2
127.0.0.1:6379> JSON.OBJLEN k1 .c.*
(error) WRONGTYPE JSON element is not an object
127.0.0.1:6379> JSON.OBJLEN k1 .d
(integer) 3
127.0.0.1:6379> JSON.OBJLEN k1 .d.*
(integer) 2
127.0.0.1:6379> JSON.OBJLEN k1 .*
(integer) 0
```

JSON.OBJKEYS

Consigue nombres clave en los valores de objeto de la ruta.

Sintaxis

```
JSON.OBJKEYS <key> [path]
```

- key (obligatorio): una clave de Valkey o Redis OSS del tipo de documento JSON.
- ruta (opcional): una ruta JSON. Toma el valor predeterminado raíz si no se proporciona.

Return

Si la ruta es de sintaxis mejorada:

- Matriz de matriz de cadenas masivas. Cada elemento es una matriz de claves de un objeto coincidente.
- Si un valor no es un objeto, su valor devuelto correspondiente es un valor vacío.

- Es nulo si la clave del documento no existe.

Si la ruta es de sintaxis restringida:

- Matriz de cadenas a granel. Cada elemento es un nombre clave del objeto.
- Si hay varios objetos seleccionados, el comando devuelve las claves del primer objeto.
- El error `WRONGTYPE` si el valor de la ruta no es un objeto.
- Es nulo si la clave del documento no existe.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{}, "b":{"a":"a"}, "c":{"a":"a", "b":"bb"}, "d":
{"a":1, "b":"b", "c":{"a":3,"b":4}}, "e":1}'
OK
127.0.0.1:6379> JSON.OBJKEYS k1 $.*
1) (empty array)
2) 1) "a"
3) 1) "a"
   2) "b"
4) 1) "a"
   2) "b"
   3) "c"
5) (empty array)
127.0.0.1:6379> JSON.OBJKEYS k1 $.d
1) 1) "a"
   2) "b"
   3) "c"
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{}, "b":{"a":"a"}, "c":{"a":"a", "b":"bb"}, "d":
{"a":1, "b":"b", "c":{"a":3,"b":4}}, "e":1}'
OK
127.0.0.1:6379> JSON.OBJKEYS k1 .*
1) "a"
127.0.0.1:6379> JSON.OBJKEYS k1 .d
1) "a"
```

```
2) "b"
3) "c"
```

JSON.RESP

Devuelve el valor de JSON en la ruta especificada en el protocolo de serialización de Valkey o Redis OSS (RESP). Si el valor es contenedor, la respuesta es una matriz RESP o matriz anidada.

- El valor nulo de JSON se asigna a la cadena masiva nula de RESP.
- Los valores boleanos JSON se asignan a las cadenas simples de RESP respectivas.
- Los números enteros se asignan a números enteros RESP.
- Los números de coma flotante doble IEEE de 64 bits se asignan a cadenas masivas RESP.
- Las cadenas JSON se asignan a RESP Bulk Strings.
- Las matrices JSON se representan como matrices RESP, donde el primer elemento es la cadena simple [, seguida de los elementos de la matriz.
- Los objetos JSON se representan como matrices RESP, donde el primer elemento es la cadena simple {, seguida de los pares clave-valor, cada uno de los cuales es una cadena masiva RESP.

Sintaxis

```
JSON.RESP <key> [path]
```

- key (obligatorio): una clave de Valkey o Redis OSS del tipo de documento JSON.
- ruta (opcional): una ruta JSON. Toma el valor predeterminado raíz si no se proporciona.

Return

Si la ruta es de sintaxis mejorada:

- Matriz de matrices. Cada elemento de la matriz representa la forma RESP del valor en una ruta.
- Matriz vacía si la clave del documento no existe.

Si la ruta es de sintaxis restringida:

- Matriz que representa la forma RESP del valor en la ruta.

- Es nulo si la clave del documento no existe.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 .
 '{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
{"street":"21 2nd Street","city":"New
 York","state":"NY","zipcode":"10021-3100"},"phoneNumbers":
[{"type":"home","number":"212 555-1234"},{"type":"office","number":"646
 555-4567"}],"children":[],"spouse":null}'
OK

127.0.0.1:6379> JSON.RESP k1 $.address
1) 1) {
   2) 1) "street"
      2) "21 2nd Street"
   3) 1) "city"
      2) "New York"
   4) 1) "state"
      2) "NY"
   5) 1) "zipcode"
      2) "10021-3100"

127.0.0.1:6379> JSON.RESP k1 $.address.*
1) "21 2nd Street"
2) "New York"
3) "NY"
4) "10021-3100"

127.0.0.1:6379> JSON.RESP k1 $.phoneNumbers
1) 1) [
   2) 1) {
      2) 1) "type"
         2) "home"
      3) 1) "number"
         2) "555 555-1234"
   3) 1) {
      2) 1) "type"
         2) "office"
      3) 1) "number"
         2) "555 555-4567"
```

```
127.0.0.1:6379> JSON.RESP k1 $.phoneNumbers[*]
1) 1) {
   2) 1) "type"
      2) "home"
   3) 1) "number"
      2) "212 555-1234"
2) 1) {
   2) 1) "type"
      2) "office"
   3) 1) "number"
      2) "555 555-4567"
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 .
 '{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
{"street":"21 2nd Street","city":"New
 York","state":"NY","zipcode":"10021-3100"},"phoneNumbers":
[{"type":"home","number":"212 555-1234"},{"type":"office","number":"646
 555-4567"}],"children":[],"spouse":null}'
OK

127.0.0.1:6379> JSON.RESP k1 .address
1) {
2) 1) "street"
   2) "21 2nd Street"
3) 1) "city"
   2) "New York"
4) 1) "state"
   2) "NY"
5) 1) "zipcode"
   2) "10021-3100"

127.0.0.1:6379> JSON.RESP k1
 1) {
 2) 1) "firstName"
    2) "John"
 3) 1) "lastName"
    2) "Smith"
 4) 1) "age"
    2) (integer) 27
```

```
 5) 1) "weight"
    2) "135.25"
 6) 1) "isAlive"
    2) true
 7) 1) "address"
    2) 1) {
       2) 1) "street"
          2) "21 2nd Street"
       3) 1) "city"
          2) "New York"
       4) 1) "state"
          2) "NY"
       5) 1) "zipcode"
          2) "10021-3100"
 8) 1) "phoneNumbers"
    2) 1) [
       2) 1) {
          2) 1) "type"
             2) "home"
          3) 1) "number"
             2) "212 555-1234"
       3) 1) {
          2) 1) "type"
             2) "office"
          3) 1) "number"
             2) "555 555-4567"
 9) 1) "children"
    2) 1) [
10) 1) "spouse"
    2) (nil)
```

JSON.SET

Establece valores JSON en la ruta.

Si la ruta de acceso llama a un miembro de objeto:

- Si el elemento principal no existe, el comando regresa unerror INEXISTENTE.
- Si el elemento principal existe pero no es un objeto, el comando regresa ERROR.
- Si el elemento principal existe y es un objeto:

- Si el miembro no existe, se anexará un miembro nuevo al objeto principal si y solo si el objeto principal es el último objeto secundario de la ruta. De lo contrario, el comando regresa un error INEXISTENTE.
- Si el miembro existe, su valor se reemplazará por el valor JSON.

Si la ruta requiere un índice de matriz:

- Si el elemento principal no existe, el comando regresa unerror INEXISTENTE.
- Si el elemento principal existe pero no es una matriz, el comando regresa ERROR.
- Si el elemento principal existe pero el índice está fuera de los límites, el comando regresa unerror FUERA DE LOS LÍMITES.
- Si el elemento principal existe y el índice es válido, el elemento se reemplazará por el nuevo valor JSON.

Si la ruta llama a un objeto o matriz, el valor (objeto o matriz) se reemplazará por el nuevo valor JSON.

Sintaxis

```
JSON.SET <key> <path> <json> [NX | XX]
```

[NX | XX] Donde puede tener 0 o 1 de [NX | XX] identificadores.

- key (obligatorio): una clave de Valkey o Redis OSS del tipo de documento JSON.
- ruta (obligatoria): una ruta JSON. Para una nueva clave, la ruta de JSON debe ser la raíz “.”.
- NX (opcional): si la ruta es la raíz, establezca el valor solo si la clave no existe. Es decir, inserte un nuevo documento. Si la ruta no es la raíz, establezca el valor solo si la ruta no existe. Es decir, inserte un valor en el documento.
- XX (opcional): si la ruta es la raíz, establezca el valor solo si existe la clave. Es decir, reemplace el documento existente. Si la ruta no es la raíz, establezca el valor solo si la ruta existe. Es decir, actualice el valor existente.

Return

- Cadena simple ‘OK’ en caso de éxito.

- Es nulo si no se cumple la condición NX o XX.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":{"a":1, "b":2, "c":3}}'
OK
127.0.0.1:6379> JSON.SET k1 $.a.* '0'
OK
127.0.0.1:6379> JSON.GET k1
"{\"a\":{\"a\":0,\"b\":0,\"c\":0}}"

127.0.0.1:6379> JSON.SET k2 . '{"a": [1,2,3,4,5]}'
OK
127.0.0.1:6379> JSON.SET k2 $.a[*] '0'
OK
127.0.0.1:6379> JSON.GET k2
"{\"a\":[0,0,0,0,0]}"
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 . '{"c":{"a":1, "b":2}, "e": [1,2,3,4,5]}'
OK
127.0.0.1:6379> JSON.SET k1 .c.a '0'
OK
127.0.0.1:6379> JSON.GET k1
"{\"c\":{\"a\":0,\"b\":2},\"e\":[1,2,3,4,5]}"
127.0.0.1:6379> JSON.SET k1 .e[-1] '0'
OK
127.0.0.1:6379> JSON.GET k1
"{\"c\":{\"a\":0,\"b\":2},\"e\":[1,2,3,4,0]}"
127.0.0.1:6379> JSON.SET k1 .e[5] '0'
(error) OUTOFBOUNDARIES Array index is out of bounds
```

JSON.STRAPPEND

Adjunta una cadena a las cadenas JSON de la ruta.

Sintaxis

```
JSON.STRAPPEND <key> [path] <json_string>
```

- key (obligatorio): una clave de Valkey o Redis OSS del tipo de documento JSON.
- ruta (opcional): una ruta JSON. Toma el valor predeterminado raíz si no se proporciona.
- json_string (obligatorio): la representación JSON de una cadena. Tenga en cuenta que se debe citar una cadena JSON. Por ejemplo: “’ejemplo de cadena‘”.

Return

Si la ruta es de sintaxis mejorada:

- Matriz de números enteros que representa la nueva longitud de la cadena en cada ruta.
- Si un valor en la ruta no es una cadena, su valor devuelto correspondiente es nulo.
- `SYNTAXERR`Error si el argumento json de entrada no es una cadena JSON válida.
- `NONEXISTENT`Error si la ruta no existe.

Si la ruta es de sintaxis restringida:

- Entero, la nueva longitud de la cadena.
- Si se seleccionan varios valores de cadena, el comando devuelve la nueva longitud de la última cadena actualizada.
- Error `WRONGTYPE` si el valor de la ruta no es una cadena.
- Error `WRONGTYPE` si el argumento json de entrada no es una cadena JSON válida.
- `NONEXISTENT`Error si la ruta no existe.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
 "b":"bb"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.STRAPPEND k1 $.a.a '"a"'
1) (integer) 2
127.0.0.1:6379> JSON.STRAPPEND k1 $.a.* '"a"'
1) (integer) 3
```

```
127.0.0.1:6379> JSON.STRAPPEND k1 $.b.* '"a"'
1) (integer) 2
2) (nil)
127.0.0.1:6379> JSON.STRAPPEND k1 $.c.* '"a"'
1) (integer) 2
2) (integer) 3
127.0.0.1:6379> JSON.STRAPPEND k1 $.c.b '"a"'
1) (integer) 4
127.0.0.1:6379> JSON.STRAPPEND k1 $.d.* '"a"'
1) (nil)
2) (integer) 2
3) (nil)
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
 "b":"bb"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.STRAPPEND k1 .a.a '"a"'
(integer) 2
127.0.0.1:6379> JSON.STRAPPEND k1 .a.* '"a"'
(integer) 3
127.0.0.1:6379> JSON.STRAPPEND k1 .b.* '"a"'
(integer) 2
127.0.0.1:6379> JSON.STRAPPEND k1 .c.* '"a"'
(integer) 3
127.0.0.1:6379> JSON.STRAPPEND k1 .c.b '"a"'
(integer) 4
127.0.0.1:6379> JSON.STRAPPEND k1 .d.* '"a"'
(integer) 2
```

JSON.STRLEN

Consigue las longitudes de los valores de cadena JSON en la ruta.

Sintaxis

```
JSON.STRLEN <key> [path]
```

- key (obligatorio): una clave de Valkey o Redis OSS del tipo de documento JSON.

- ruta (opcional): una ruta JSON. Toma el valor predeterminado raíz si no se proporciona.

Return

Si la ruta es de sintaxis mejorada:

- Matriz de números enteros que representa la longitud del valor de cadena en cada ruta.
- Si un valor no es una cadena, su valor devuelto correspondiente es nulo.
- Es nulo si la clave del documento no existe.

Si la ruta es de sintaxis restringida:

- Entero, la longitud de la cadena.
- Si se seleccionan varios valores de cadena, el comando devuelve la longitud de la primera cadena.
- Error `WRONGTYPE` si el valor de la ruta no es una cadena.
- Error `NONEXISTENT` si la ruta no existe.
- Es nulo si la clave del documento no existe.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
 "b":"bb"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.STRLEN k1 $.a.a
1) (integer) 1
127.0.0.1:6379> JSON.STRLEN k1 $.a.*
1) (integer) 1
127.0.0.1:6379> JSON.STRLEN k1 $.c.*
1) (integer) 1
2) (integer) 2
127.0.0.1:6379> JSON.STRLEN k1 $.c.b
1) (integer) 2
127.0.0.1:6379> JSON.STRLEN k1 $.d.*
1) (nil)
2) (integer) 1
3) (nil)
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
 "b":"bb"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.STRLEN k1 .a.a
(integer) 1
127.0.0.1:6379> JSON.STRLEN k1 .a.*
(integer) 1
127.0.0.1:6379> JSON.STRLEN k1 .c.*
(integer) 1
127.0.0.1:6379> JSON.STRLEN k1 .c.b
(integer) 2
127.0.0.1:6379> JSON.STRLEN k1 .d.*
(integer) 1
```

JSON.TOGGLE

Alterna los valores boleanos entre verdadero y falso en el camino.

Sintaxis

```
JSON.TOGGLE <key> [path]
```

- key (obligatorio): una clave de Valkey o Redis OSS del tipo de documento JSON.
- ruta (opcional): una ruta JSON. Toma el valor predeterminado raíz si no se proporciona.

Return

Si la ruta es de sintaxis mejorada:

- Matriz de números enteros (0 - falso, 1 - verdadero) que representa el valor boleano resultante en cada ruta.
- Si un valor no es un valor boleano, su valor devuelto correspondiente es nulo.
- `NONEXISTENT` si la clave del documento no existe.

Si la ruta es de sintaxis restringida:

- Cadena (“verdadero”/”falso”) que representa el resultantevalor boleano.
- `NONEXISTENT` si la clave del documento no existe.
- Error `WRONGTYPE` si el valor de la ruta no es un valor boleano.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":true, "b":false, "c":1, "d":null, "e":"foo", "f":
[], "g":{}}'
OK
127.0.0.1:6379> JSON.TOGGLE k1 $.*
1) (integer) 0
2) (integer) 1
3) (nil)
4) (nil)
5) (nil)
6) (nil)
7) (nil)
127.0.0.1:6379> JSON.TOGGLE k1 $.*
1) (integer) 1
2) (integer) 0
3) (nil)
4) (nil)
5) (nil)
6) (nil)
7) (nil)
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 . true
OK
127.0.0.1:6379> JSON.TOGGLE k1
"false"
127.0.0.1:6379> JSON.TOGGLE k1
"true"

127.0.0.1:6379> JSON.SET k2 . '{"isAvailable": false}'
OK
127.0.0.1:6379> JSON.TOGGLE k2 .isAvailable
"true"
```

```
127.0.0.1:6379> JSON.TOGGLE k2 .isAvailable
"false"
```

JSON.TYPE

Informa el tipo de valores en la ruta dada.

Sintaxis

```
JSON.TYPE <key> [path]
```

- key (obligatorio): una clave de Valkey o Redis OSS del tipo de documento JSON.
- ruta (opcional): una ruta JSON. Toma el valor predeterminado raíz si no se proporciona.

Return

Si la ruta es de sintaxis mejorada:

- Matriz de cadenas que representa el tipo de valor en cada ruta. El tipo es uno de {"nulo", "boleano", "cadena", "número", "entero", "objeto" y "matriz"}.
- Si no existe una ruta, su valor de retorno correspondiente es nulo.
- Matriz vacía si la clave del documento no existe.

Si la ruta es de sintaxis restringida:

- Cadena, tipo del valor
- Es nulo si la clave del documento no existe.
- Es nulo si la ruta JSON no es válida o no existe.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 . '[1, 2.3, "foo", true, null, {}, []]'
OK
127.0.0.1:6379> JSON.TYPE k1 $[*]
1) integer
```

```
2) number
3) string
4) boolean
5) null
6) object
7) array
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 .
 '{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
{"street":"21 2nd Street","city":"New
 York","state":"NY","zipcode":"10021-3100"},"phoneNumbers":
[{"type":"home","number":"212 555-1234"},{"type":"office","number":"646
 555-4567"}],"children":[],"spouse":null}'
OK
127.0.0.1:6379> JSON.TYPE k1
object
127.0.0.1:6379> JSON.TYPE k1 .children
array
127.0.0.1:6379> JSON.TYPE k1 .firstName
string
127.0.0.1:6379> JSON.TYPE k1 .age
integer
127.0.0.1:6379> JSON.TYPE k1 .weight
number
127.0.0.1:6379> JSON.TYPE k1 .isAlive
boolean
127.0.0.1:6379> JSON.TYPE k1 .spouse
null
```

Etiquetado de los recursos de ElastiCache

Para ayudarlo a administrar sus clústeres y otros recursos de ElastiCache, puede asignar sus propios metadatos a cada recurso en forma de etiquetas. Las etiquetas le permiten clasificar los recursos de AWS de diversas maneras, por ejemplo, según su finalidad, propietario o entorno. Esto es útil cuando tiene muchos recursos del mismo tipo: puede identificar rápidamente un recurso específico en función de las etiquetas que le haya asignado. En este tema se describe qué son las etiquetas y cómo crearlas.

> **Warning**
>
> Como práctica recomendada, no debe incluir datos confidenciales en las etiquetas.

Conceptos básicos de etiquetas

Una etiqueta es una marca que se asigna a un recurso de AWS. Cada etiqueta está formada por una clave y un valor opcional, ambos definidos por el usuario. Las etiquetas permiten clasificar los recursos de AWS de diversas maneras, por ejemplo, según finalidad o propietario. Por ejemplo, podría definir un conjunto de etiquetas para los clústeres de ElastiCache de su cuenta que lo ayude a realizar un seguimiento del propietario y el grupo de usuario de cada instancia.

Recomendamos que idee un conjunto de claves de etiqueta que cumpla sus necesidades para cada tipo de recurso. Mediante el uso de un conjunto coherente de claves de etiquetas, podrá administrar los recursos más fácilmente. Puede buscar y filtrar los recursos en función de las etiquetas que agregue. Para obtener más información acerca de cómo implementar una estrategia eficaz de etiquetado de recursos, consulte el documento técnico de Prácticas recomendadas de etiquetado de AWS.

Las etiquetas no tienen significado semántico para ElastiCache, por lo que se interpretan estrictamente como cadenas de caracteres. Además, las etiquetas no se asignan a los recursos automáticamente. Puede editar las claves y los valores de las etiquetas y también puede eliminar etiquetas de un recurso en cualquier momento. Puede establecer el valor de una etiqueta en `null`. Si añade una etiqueta con la misma clave que una etiqueta existente en ese recurso, el nuevo valor sobrescribirá al antiguo. Si elimina un recurso, también se eliminará cualquier etiqueta asignada a dicho recurso. Además, si agrega o elimina etiquetas en un grupo de reproducción, todos los nodos de ese grupo de reproducción también se agregarán o eliminarán sus etiquetas.

Puede trabajar con etiquetas mediante la Consola de administración de AWS, la AWS CLI y la API de ElastiCache.

Si utiliza IAM, puede controlar qué usuarios de su cuenta de AWS tienen permiso para crear, editar o eliminar etiquetas. Para obtener más información, consulte Permisos de nivel de recursos.

Recursos que se pueden etiquetar

Puede etiquetar la mayoría de los recursos de ElastiCache que ya existen en la cuenta. La siguiente tabla enumera los recursos que admiten etiquetas. Si utiliza la Consola de administración de AWS,

puede aplicar etiquetas a recursos a través del Editor de etiquetas. Algunas pantallas de recursos permiten especificar etiquetas para un recurso al crear dicho recurso; por ejemplo, una etiqueta con una clave de Name (Nombre) y un valor que especifique. En la mayoría de los casos, la consola aplica las etiquetas inmediatamente después de crear el recurso (y no durante la creación del mismo). La consola puede organizar los recursos según la etiqueta de Name (Nombre), si bien dicha etiqueta no tiene significado semántico para el servicio de ElastiCache.

Además, algunas acciones de creación de recursos le permiten especificar etiquetas para un recurso al crear dicho recurso. Si no se pueden aplicar etiquetas durante la creación del recurso, el proceso de creación del recurso se revierte. Esto garantiza que los recursos se creen con etiquetas o, de lo contrario, no se creen y que ningún recurso se quede jamás sin etiquetar. Al etiquetar los recursos en el momento de su creación, se eliminar la necesidad de ejecutar scripts de etiquetado personalizados tras la creación del recurso.

Si utiliza la API de Amazon ElastiCache, la AWS CLI o un SDK de AWS, puede aplicar etiquetas mediante el parámetro `Tags` en la acción de la API de ElastiCache pertinente. Son los siguientes:

- `CreateServerlessCache`
- `CreateCacheCluster`
- `CreateReplicationGroup`
- `CopyServerlessCacheSnapshot`
- `CopySnapshot`
- `CreateCacheParameterGroup`
- `CreateCacheSecurityGroup`
- `CreateCacheSubnetGroup`
- `CreateServerlessCacheSnapshot`
- `CreateSnapshot`
- `CreateUserGroup`
- `CreateUser`
- `PurchaseReservedCacheNodesOffering`

En la siguiente tabla se describen los recursos de ElastiCache que se pueden etiquetar y aquellos que se pueden etiquetar en el momento de su creación con la API de ElastiCache, la AWS CLI o un SDK de AWS.

Compatibilidad con el etiquetado de recursos de ElastiCache

Admite etiquetas	Admite el etiquetad o durante la creación
Sí	Sí
Sí	Sí
Sí	Sí
Sí	Sí
Sí	Sí
Sí	Sí
Sí	Sí
Sí	Sí
Sí	Sí
Sí	Sí
Sí	Sí

> **Note**
>
> No puede etiquetar almacenes de datos globales.

En las políticas de IAM, puede aplicar permisos de nivel de recursos basados en etiquetas a las acciones de la API de ElastiCache que admitan el etiquetado durante la creación para implementar un control pormenorizado de los usuarios y los grupos que pueden etiquetar recursos durante la creación. Sus recursos se encuentran debidamente protegidos de las etiquetas de creación que se aplican de inmediato a los recursos. Por lo tanto, cualquier permiso de nivel de recursos basado en etiquetas que controle la utilización de recursos es efectivo de inmediato. Se puede realizar un seguimiento y un registro más precisos de los recursos. Puede establecer el etiquetado obligatorio de los nuevos recursos y controlar qué claves y valores de etiquetas se usan en ellos.

Para obtener más información, consulte Ejemplos de etiquetado de recursos.

A fin de obtener más información sobre el etiquetado de recursos para facturación, consulte Monitoreo de costos con etiquetas de asignación de costos.

Etiquetado de cachés e instantáneas

Las siguientes reglas se aplican al etiquetado como parte de las operaciones de solicitud:

- CreateReplicationGroup:
  - Si los parámetros `--primary-cluster-id` y `--tags` se incluyen en la solicitud, las etiquetas de la solicitud se agregarán al grupo de replicación y se propagan a todos los clústeres del grupo de replicación. Si el clúster principal tiene etiquetas, estas se sobrescribirán con las etiquetas de la solicitud para tener etiquetas coherentes en todos los nodos.

    Si no hay etiquetas de la solicitud, las etiquetas de clúster principal se agregarán al grupo de replicación y se propagarán a todos los clústeres.
  - Si se suministra el `--snapshot-name` o el `--serverless-cache-snapshot-name`:

    Si se incluyen etiquetas en la solicitud, el grupo de réplica solo se etiquetará con esas etiquetas. Si no se incluyen etiquetas en la solicitud, las etiquetas de la instantánea se agregarán al grupo de réplica.
  - Si se proporciona el `--global-replication-group-id`:

Si se incluyen etiquetas en la solicitud, las etiquetas de la solicitud se agregarán al grupo de replicación y se propagarán a todos los clústeres de caché.

- CreateCacheCluster :
  - Si se proporciona el `--replication-group-id`:

    Si se incluyen etiquetas en la solicitud, solo se le aplicarán esas etiquetas al clúster. Si no se incluyen etiquetas en la solicitud, el clúster heredará las etiquetas del grupo de replicación en lugar de las etiquetas del clúster principal.
  - Si se proporciona el `--snapshot-name`:

    Si se incluyen etiquetas en la solicitud, solo se le aplicarán esas etiquetas al clúster. Si no se incluyen etiquetas en la solicitud, las etiquetas de la instantánea se agregarán al clúster.
- CreateServerlessCache:
  - Si se incluyen etiquetas en la solicitud, solo las etiquetas de solicitud se agregarán al caché sin servidor.
- CreateSnapshot :
  - Si se proporciona el `--replication-group-id`:

    Si no se incluyen etiquetas en la solicitud, las etiquetas de solicitud se agregarán a la instantánea. Si no se incluyen etiquetas en la solicitud, las etiquetas del grupo de reproducción se agregarán a la instantánea.
  - Si se proporciona el `--cache-cluster-id`:

    Si no se incluyen etiquetas en la solicitud, las etiquetas de solicitud se agregarán a la instantánea. Si no se incluyen etiquetas en la solicitud, las etiquetas del clúster se agregarán a la instantánea.
  - Para las instantáneas automáticas:

    Las etiquetas se propagarán desde las etiquetas de grupo de reproducción.
- CreateServerlessCacheSnapshot:
  - Si se incluyen etiquetas en la solicitud, solo las etiquetas de solicitud se agregarán a la instantánea de la caché sin servidor.
- CopySnapshot :

  - Si no se incluyen etiquetas en la solicitud, las etiquetas de solicitud se agregarán a la instantánea. Si no se incluyen etiquetas en la solicitud, las etiquetas de la instantánea fuente se agregarán a la instantánea copiada.
- CopyServerlessCacheSnapshot:
  - Si se incluyen etiquetas en la solicitud, solo las etiquetas de solicitud se agregarán a la instantánea de la caché sin servidor.
- AddTagsToResource y RemoveTagsFromResource:
  - Las etiquetas se agregarán o quitarán del grupo de reproducción y la acción se propagará a todos los clústeres del grupo de reproducción.

> **Note**
>
> AddTagsToResource y RemoveTagsFromResource no se pueden utilizar para los grupos de seguridad y parámetros predeterminados.

- IncreaseReplicaCount y ModifyReplicationGroupShardConfiguration:
  - Todos los clústeres nuevos agregados al grupo de réplica tendrán las mismas etiquetas aplicadas que el grupo de réplica.

Restricciones de las etiquetas

Se aplican las siguientes restricciones básicas a las etiquetas:

- Número máximo de etiquetas por recurso: 50
- Para cada recurso, cada clave de etiqueta debe ser única y solo puede tener un valor.
- Longitud máxima de la clave: 128 caracteres Unicode en UTF-8.
- Longitud máxima del valor: 256 caracteres Unicode en UTF-8.
- Si bien ElastiCache admite la utilización de cualquier carácter en sus etiquetas, otros servicios pueden ser restrictivos. Los caracteres permitidos en los servicios son: letras, números y espacios representables en UTF-8, además de los siguientes caracteres: + - = . _ : / @
- Las claves y los valores de las etiquetas distinguen entre mayúsculas y minúsculas.
- El prefijo `aws:` se reserva para uso de AWS. Si la etiqueta tiene una clave de etiqueta con este prefijo, no puede editar ni eliminar la clave o el valor de la etiqueta. Las etiquetas que tengan el prefijo `aws:` no cuentan para el límite de etiquetas por recurso.

No puede finalizar, detener ni eliminar un recurso basado únicamente en sus etiquetas; debe especificar el identificador del recurso. Por ejemplo, para eliminar instantáneas que etiquetó con una clave de etiqueta llamada `DeleteMe`, debe utilizar la acción `DeleteSnapshot` con los identificadores del recurso de las instantáneas, como `snap-1234567890abcdef0`.

Para obtener más información sobre los recursos de ElastiCache que puede etiquetar, consulte Recursos que se pueden etiquetar.

Ejemplos de etiquetado de recursos

- Creación de una caché sin servidor mediante etiquetas. En este ejemplo, se utiliza Memcached como motor.

```
aws elasticache create-serverless-cache \
    --serverless-cache-name CacheName \
    --engine memcached
    --tags Key="Cost Center", Value="1110001" Key="project",Value="XYZ"
```

- Adición de etiquetas a una caché sin servidor

```
aws elasticache add-tags-to-resource \
--resource-name arn:aws:elasticache:us-east-1:111111222233:serverlesscache:my-cache \
--tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- Adición de etiquetas a un grupo de reproducción.

```
aws elasticache add-tags-to-resource \
--resource-name arn:aws:elasticache:us-east-1:111111222233:replicationgroup:my-rg \
--tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- Creación de un clúster de caché mediante etiquetas.

```
aws elasticache create-cache-cluster \
--cluster-id testing-tags \
--cluster-description cluster-test \
--cache-subnet-group-name test \
--cache-node-type cache.t2.micro \
--engine valkey \
--tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- Creación de un clúster de caché mediante etiquetas. En este ejemplo, se utiliza Redis como motor.

```
aws elasticache create-cache-cluster \
--cluster-id testing-tags \
--cluster-description cluster-test \
--cache-subnet-group-name test \
--cache-node-type cache.t2.micro \
--engine valkey \
--tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- Creación de una instantánea sin servidor con etiquetas. En este ejemplo, se utiliza Memcached como motor.

```
aws elasticache create-serverless-cache-snapshot \
--serverless-cache-name testing-tags \
--serverless-cache-snapshot-name bkp-testing-tags-scs \
--tags Key="work",Value="foo"
```

- Creación de una instantánea con etiquetas.

  Actualmente, las instantáneas solo están disponibles para Redis. En este caso, si agrega etiquetas a la solicitud, incluso si el grupo de reproducción contiene etiquetas, la instantánea solo recibirá las etiquetas de la solicitud.

```
aws elasticache create-snapshot \
--replication-group-id testing-tags \
--snapshot-name bkp-testing-tags-rg \
--tags Key="work",Value="foo"
```

Ejemplos de políticas de control de acceso basadas en etiquetas

1. Permitir la acción `AddTagsToResource` a un clúster solo si el clúster tiene la etiqueta Project=XYZ.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
```

```
            "Action": "elasticache:AddTagsToResource",
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Project": "XYZ"
                }
            }
        }
    ]
}
```

2. Permitir la acción `RemoveTagsFromResource` de un grupo de reproducción si contiene las etiquetas Project y Service y las claves son diferentes de Project y Service.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "elasticache:RemoveTagsFromResource",
            "Resource": [
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Service": "Elasticache",
                    "aws:ResourceTag/Project": "XYZ"
                },
                "ForAnyValue:StringNotEqualsIgnoreCase": {
                    "aws:TagKeys": [
                        "Project",
                        "Service"
                    ]
                }
            }
        }
    ]
}
```

3. Permitir `AddTagsToResource` a cualquier recurso solo si las etiquetas son diferentes de Project y Service.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "elasticache:AddTagsToResource",
            "Resource": [
                "arn:aws:elasticache:*:*:*:*"
            ],
            "Condition": {
                "ForAnyValue:StringNotEqualsIgnoreCase": {
                    "aws:TagKeys": [
                        "Service",
                        "Project"
                    ]
                }
            }
        }
    ]
}
```

4. Denegar la acción `CreateReplicationGroup` si la solicitud tiene `Tag Project=Foo`.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "elasticache:CreateReplicationGroup",
            "Resource": [
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/Project": "Foo"
                }
            }
```

```
        }
    ]
}
```

5. Denegar la acción `CopySnapshot` si la instantánea fuente tiene la etiqueta Project=XYZ y la etiqueta de la solicitud es Service=Elasticache.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "elasticache:CopySnapshot",
            "Resource": [
                "arn:aws:elasticache:*:*:snapshot:*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Project": "XYZ",
                    "aws:RequestTag/Service": "Elasticache"
                }
            }
        }
    ]
}
```

6. Denegar la acción `CreateCacheCluster` si la etiqueta de solicitud `Project` falta o no es igual a `Dev`, `QA` o `Prod`.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
          {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*",
```

```
                "arn:aws:elasticache:*:*:securitygroup:*",
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ]
        },
        {
            "Effect": "Deny",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*"
            ],
            "Condition": {
                "Null": {
                    "aws:RequestTag/Project": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:AddTagsToResource"
            ],
            "Resource": "arn:aws:elasticache:*:*:cluster:*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/Project": [
                        "Dev",
                        "Prod",
                        "QA"
                    ]
                }
            }
        }
    ]
}
```

Para obtener información relacionada acerca de las claves de condición, consulte Uso de claves de condición.

Monitoreo de costos con etiquetas de asignación de costos

Al agregar etiquetas de asignación de costos a sus recursos en Amazon ElastiCache, puede realizar un seguimiento de los costos mediante el agrupamiento de los gastos en sus facturas por valores de etiqueta de recursos.

Las etiquetas de asignación de costos de ElastiCache son pares clave-valor que el usuario define y asocia a un recurso de ElastiCache. Las claves y los valores distinguen entre mayúsculas y minúsculas. Puede utilizar una clave de etiqueta para definir una categoría y el valor de la etiqueta puede ser un elemento dentro de esa categoría. Por ejemplo, puede definir una clave de etiqueta `CostCenter` y un valor de etiqueta `10010` para indicar que el recurso va asignado al centro de costos 10010. También puede usar etiquetas para designar recursos para pruebas o para producción a través de una clave como `Environment` y valores como `test` o `production`. Se recomienda utilizar un conjunto coherente de claves de etiqueta que facilite el seguimiento de los costos asociados a los recursos.

Utilice las etiquetas de asignación de costos para organizar la factura de AWS de modo que refleje su propia estructura de costos. Para ello, inscríbase para obtener una factura de la cuenta de AWS que incluya valores de clave de etiquetas. A continuación, para ver los costos de los recursos combinados, organice la información de facturación de acuerdo con los recursos con los mismos valores de clave de etiquetas. Por ejemplo, puede etiquetar varios recursos con un nombre de aplicación específico y luego organizar su información de facturación para ver los costos totales de la aplicación en distintos servicios.

También puede combinar etiquetas para realizar un seguimiento de los costos con un mayor nivel de detalle. Por ejemplo, para realizar un seguimiento de los costos de su servicio por región, puede utilizar las claves de etiqueta `Service` y `Region`. En un recurso podría tener los valores `ElastiCache` y `Asia Pacific (Singapore)` y en otro recurso, los valores `ElastiCache` y `Europe (Frankfurt)`. A continuación, puede ver el total de costos de ElastiCache desglosados por región. Para obtener más información, consulte Uso de etiquetas de asignación de costos en la Guía del usuario de AWS Billing.

Puede añadir etiquetas de asignación de costos de ElastiCache a los clústeres basados en nodos de ElastiCache. Al agregar, enumerar, modificar, copiar o quitar una etiqueta, la operación se aplica únicamente al clúster especificado.

Características de las etiquetas de asignación de costos de ElastiCache

- Las etiquetas de asignación de costos se aplican a recursos de ElastiCache especificados en operaciones de la API y de la CLI como ARN. El tipo de recurso será "clúster".

  Ejemplo de ARN: `arn:aws:elasticache:<region>:<customer-id>:<resource-type>:<resource-name>`

  Ejemplo de ARN: `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

- La clave de la etiqueta es el nombre obligatorio de la etiqueta. El valor de cadena de la clave puede tener una longitud de entre 1 y 128 caracteres Unicode y no puede llevar el prefijo `aws:`. La cadena solo puede contener un conjunto Unicode de letras, dígitos, espacios en blanco, guiones bajos (_), puntos (.), dos puntos (:), barras oblicuas (\ ), signos de igual (=), signos de suma (+), guiones (-) o signos de arroba (@).

- El valor de etiqueta es la parte opcional de la etiqueta. El valor de cadena del valor puede tener una longitud de entre 1 y 256 caracteres Unicode y no puede llevar el prefijo `aws:`. La cadena solo puede contener un conjunto Unicode de letras, dígitos, espacios en blanco, guiones bajos (_), puntos (.), dos puntos (:), barras oblicuas (\ ), signos de igual (=), signos de suma (+), guiones (-) o signos de arroba (@).

- Cada recurso de ElastiCache puede tener un máximo de 50 etiquetas.

- Los valores no deben ser únicos dentro de un conjunto de etiquetas. Por ejemplo, puede disponer de un conjunto de etiquetas donde las claves `Service` y `Application` tienen el valor `ElastiCache`.

AWS no aplica ningún significado semántico a las etiquetas. Las etiquetas se interpretan estrictamente como cadenas de caracteres. AWS no establece de forma automática ninguna etiqueta en ningún recurso de ElastiCache.

Administración de etiquetas de asignación de costos mediante la AWS CLI

Puede utilizar la AWS CLI para agregar, modificar o quitar etiquetas de asignación de costos.

Las etiquetas de asignación de costos se aplican a los clústeres de ElastiCache. El clúster que se va a etiquetar se especifica mediante un ARN (nombre de recurso de Amazon).

Ejemplo de ARN: `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

Temas

- Enumeración de etiquetas mediante la AWS CLI
- Adición de etiquetas mediante la AWS CLI
- Modificación de etiquetas mediante la AWS CLI
- Eliminación de etiquetas mediante la AWS CLI

Enumeración de etiquetas mediante la AWS CLI

Puede utilizar la AWS CLI para enumerar etiquetas en un recurso de ElastiCache existente mediante la operación list-tags-for-resource.

El código siguiente utiliza la AWS CLI para obtener una lista de las etiquetas en el clúster de Memcached `my-cluster` en la región us-west-2.

Para Linux, macOS o Unix:

```
aws elasticache list-tags-for-resource \
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
```

Para Windows:

```
aws elasticache list-tags-for-resource ^
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
```

El código siguiente utiliza la AWS CLI para obtener una lista de las etiquetas del nodo de Valkey o Redis OSS `my-cluster-001` en el clúster `my-cluster` de la región us-west-2.

Para Linux, macOS o Unix:

```
aws elasticache list-tags-for-resource \
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
```

Para Windows:

```
aws elasticache list-tags-for-resource ^
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
```

La salida de esta operación se parecerá a lo siguiente, una lista de todas las etiquetas en el recurso.

```
{
   "TagList": [
      {
         "Value": "10110",
         "Key": "CostCenter"
      },
      {
         "Value": "EC2",
         "Key": "Service"
      }
   ]
}
```

Si no hay etiquetas en el recurso, la salida será una TagList vacía.

```
{
   "TagList": []
}
```

Para obtener más información, consulte la AWS CLI para list-tags-for-resource de ElastiCache.

Adición de etiquetas mediante la AWS CLI

Puede utilizar la AWS CLI para agregar etiquetas a un recurso de ElastiCache existente mediante la operación add-tags-to-resource de la CLI. Si la clave de etiqueta no existe en el recurso, la clave y el valor se añadirán a los recursos. Si la clave ya existe en el recurso, el valor asociado a dicha clave se actualizará al nuevo valor.

El siguiente código utiliza la AWS CLI para añadir las claves `Service` y `Region` con los valores `elasticache` y `us-west-2` respectivamente en el nodo `my-cluster-001` del clúster `my-cluster` en la región us-west-2.

Memcached

Para Linux, macOS o Unix:

```
aws elasticache add-tags-to-resource \
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster \
 --tags Key=Service,Value=elasticache \
        Key=Region,Value=us-west-2
```

Para Windows:

```
aws elasticache add-tags-to-resource ^
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster ^
 --tags Key=Service,Value=elasticache ^
        Key=Region,Value=us-west-2
```

Redis

Para Linux, macOS o Unix:

```
aws elasticache add-tags-to-resource \
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 \
 --tags Key=Service,Value=elasticache \
        Key=Region,Value=us-west-2
```

Para Windows:

```
aws elasticache add-tags-to-resource ^
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 ^
 --tags Key=Service,Value=elasticache ^
        Key=Region,Value=us-west-2
```

Tras la operación, la salida de esta operación se parecerá a lo siguiente, una lista de todas las etiquetas en el recurso.

```
{
   "TagList": [
      {
         "Value": "elasticache",
         "Key": "Service"
      },
      {
         "Value": "us-west-2",
         "Key": "Region"
      }
```

```
    ]
}
```

Para obtener más información, consulte la AWS CLI para add-tags-to-resource de ElastiCache.

También puede utilizar la AWS CLI para agregar etiquetas a un clúster al crear un nuevo clúster utilizando la operación create-cache-cluster. No se pueden agregar etiquetas al crear un clúster mediante la Management Console de ElastiCache. Una vez creado el clúster, puede utilizar la consola para agregar etiquetas al clúster.

Modificación de etiquetas mediante la AWS CLI

Puede utilizar la AWS CLI para modificar las etiquetas de un clúster de ElastiCache.

Para modificar las etiquetas:

- Use add-tags-to-resource para agregar una etiqueta y un valor nuevos o para cambiar el valor asociado a una etiqueta existente.
- Use remove-tags-from-resource para quitar etiquetas especificadas del recurso.

La salida de cualquier operación será una lista de las etiquetas y sus valores en el clúster especificado.

Eliminación de etiquetas mediante la AWS CLI

Puede utilizar la AWS CLI para quitar etiquetas de un clúster de ElastiCache for Memcached mediante la operación remove-tags-from-resource.

El código siguiente utiliza la AWS CLI para quitar las etiquetas con las claves `Service` y `Region` del nodo `my-cluster-001` en el clúster `my-cluster` de la región us-west-2.

Para Linux, macOS o Unix:

```
aws elasticache remove-tags-from-resource \
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster \
 --tag-keys PM Service
```

Para Windows:

```
aws elasticache remove-tags-from-resource ^
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster ^
```

```
 --tag-keys PM Service
```

En Redis OSS, el código siguiente utiliza la AWS CLI para quitar las etiquetas con las claves `Service` y `Region` del nodo `my-cluster-001` en el clúster `my-cluster` de la región us-west-2.

Para Linux, macOS o Unix:

```
aws elasticache remove-tags-from-resource \
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 \
 --tag-keys PM Service
```

Para Windows:

```
aws elasticache remove-tags-from-resource ^
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 ^
 --tag-keys PM Service
```

Tras la operación, la salida de esta operación se parecerá a lo siguiente, una lista de todas las etiquetas en el recurso.

```
{
   "TagList": []
}
```

Para obtener más información, consulte la AWS CLI para remove-tags-from-resource de ElastiCache.

Administración de etiquetas de asignación de costos mediante la API de ElastiCache

Puede utilizar la API de ElastiCache para agregar, modificar o quitar etiquetas de asignación de costos.

Las etiquetas de asignación de costos se aplican a los clústeres de ElastiCache for Memcached. El clúster que se va a etiquetar se especifica mediante un ARN (nombre de recurso de Amazon).

Ejemplo de ARN: `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

Temas

- Enumeración de etiquetas mediante la API de ElastiCache

- Adición de etiquetas mediante la API de ElastiCache
- Modificación de etiquetas con la API de ElastiCache
- Eliminación de etiquetas con la API de ElastiCache

Enumeración de etiquetas mediante la API de ElastiCache

Puede utilizar la API de ElastiCache para enumerar etiquetas en un recurso existente mediante la operación ListTagsForResource.

En Memcached, el siguiente código utiliza la API de ElastiCache para enumerar las etiquetas en el recurso `my-cluster` en la región us-west-2.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=ListTagsForResource
   &ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Version=2015-02-02
   &Timestamp=20150202T192317Z
   &X-Amz-Credential=<credential>
```

En Redis OSS, el siguiente código utiliza la API de ElastiCache para enumerar las etiquetas en el recurso `my-cluster-001` en la región us-west-2.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=ListTagsForResource
   &ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Version=2015-02-02
   &Timestamp=20150202T192317Z
   &X-Amz-Credential=<credential>
```

Adición de etiquetas mediante la API de ElastiCache

Puede utilizar la API de ElastiCache para agregar etiquetas a un clúster de ElastiCache existente mediante la operación AddTagsToResource. Si la clave de etiqueta no existe en el recurso, la clave y el valor se añadirán a los recursos. Si la clave ya existe en el recurso, el valor asociado a dicha clave se actualizará al nuevo valor.

El siguiente código utiliza la API de ElastiCache para agregar las claves `Service` y `Region` con los valores `elasticache` y `us-west-2` respectivamente. En el caso de Memcached, esto se aplica al recurso `my-cluster`. En el caso de Redis OSS, esto se aplica al recurso `my-cluster-001` de la región us-west-2.

Memcached

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=AddTagsToResource
   &ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Tags.member.1.Key=Service
   &Tags.member.1.Value=elasticache
   &Tags.member.2.Key=Region
   &Tags.member.2.Value=us-west-2
   &Version=2015-02-02
   &Timestamp=20150202T192317Z
   &X-Amz-Credential=<credential>
```

Redis

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=AddTagsToResource
   &ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Tags.member.1.Key=Service
   &Tags.member.1.Value=elasticache
   &Tags.member.2.Key=Region
   &Tags.member.2.Value=us-west-2
   &Version=2015-02-02
   &Timestamp=20150202T192317Z
   &X-Amz-Credential=<credential>
```

Para obtener más información, consulte AddTagsToResource en la Referencia de la API de Amazon ElastiCache.

Modificación de etiquetas con la API de ElastiCache

Puede utilizar la API de ElastiCache para modificar las etiquetas de un clúster de ElastiCache.

Para modificar el valor de una etiqueta:

- Use la operación AddTagsToResource para agregar una etiqueta y un valor nuevos o para cambiar el valor de una etiqueta existente.
- Use RemoveTagsFromResource para quitar etiquetas del recurso.

La salida de cualquier operación será una lista de las etiquetas y sus valores en el recurso especificado.

Use RemoveTagsFromResource para quitar etiquetas del recurso.

Eliminación de etiquetas con la API de ElastiCache

Puede utilizar la API de ElastiCache para quitar etiquetas de un clúster de ElastiCache for Memcached mediante la operación RemoveTagsFromResource.

El siguiente código utiliza la API de ElastiCache para quitar las etiquetas con las claves `Service` y `Region` del nodo `my-cluster-001` en el clúster `my-cluster` de la región us-west-2.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=RemoveTagsFromResource
   &ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &TagKeys.member.1=Service
   &TagKeys.member.2=Region
   &Version=2015-02-02
   &Timestamp=20150202T192317Z
   &X-Amz-Credential=<credential>
```

Uso del enfoque Well-Architected de Amazon ElastiCache

En esta sección se describe el enfoque Well-Architected de Amazon ElastiCache, una colección de principios de diseño y directrices para diseñar cargas de trabajo de ElastiCache con una buena arquitectura.

- El enfoque de ElastiCache se suma al Marco de AWS Well-Architected.
- Cada pilar tiene un conjunto de preguntas para ayudar a iniciar el debate sobre la arquitectura de ElastiCache.

  - Cada pregunta tiene una serie de prácticas principales junto con sus puntuaciones para la presentación de informes.
    - Obligatorio: necesario antes de pasar a producción (su ausencia representa un alto riesgo)
    - Lo mejor: el mejor estado posible en el que podría estar un cliente
    - Bueno: lo que recomendamos a los clientes que hagan (si no hay un riesgo medio)
- Terminología de Well-Architected
  - Componente: código, configuración y recursos de AWS que, en conjunto, cumplen con un requisito. Los componentes interactúan con otros componentes y, a menudo, equivalen a un servicio en las arquitecturas de microservicios.
  - Carga de trabajo: conjunto de componentes que, en conjunto, ofrecen valor empresarial. Algunos ejemplos de cargas de trabajo son sitios web de marketing, sitios web de comercio electrónico, el backend de una aplicación móvil, plataformas de análisis, etc.

> **Note**
>
> Esta guía no se ha actualizado para incluir información sobre el almacenamiento en caché de ElastiCache sin servidor y el nuevo motor de Valkey.

Temas

- Pilar de excelencia operativa del enfoque Well-Architected de Amazon ElastiCache
- Pilar de seguridad del enfoque Well-Architected de Amazon ElastiCache
- Pilar de fiabilidad del enfoque Well-Architected de Amazon ElastiCache
- Pilar de eficiencia del rendimiento del enfoque Well-Architected de Amazon ElastiCache
- Pilar de optimización de costos del enfoque Well-Architected de Amazon ElastiCache

Pilar de excelencia operativa del enfoque Well-Architected de Amazon ElastiCache

El pilar de la excelencia operativa se centra en el funcionamiento y la supervisión de los sistemas para ofrecer valor empresarial, y en la mejora continua de los procesos y los procedimientos. Los temas clave incluyen la automatización de los cambios, la respuesta a los eventos y la definición de estándares para administrar las operaciones diarias.

Temas

- OE 1: ¿Cómo entiende y responde a las alertas y eventos activados por su clúster de ElastiCache?
- OE 2: ¿Cuándo y cómo se escalan los clústeres de ElastiCache existentes?
- OE 3: ¿Cómo se administran los recursos del clúster de ElastiCache y se mantiene actualizado?
- OE 4: ¿Cómo se administran las conexiones de los clientes a sus clústeres de ElastiCache?
- OE 5: ¿Cómo se implementan los componentes de ElastiCache para una carga de trabajo?
- OE 6: ¿Cómo se planifican y mitigan los errores?
- OE 7: ¿Cómo se solucionan los problemas de los eventos del motor de Valkey o Redis OSS?

OE 1: ¿Cómo entiende y responde a las alertas y eventos activados por su clúster de ElastiCache?

Introducción a nivel de pregunta: Si utiliza clústeres de ElastiCache, puede recibir notificaciones y alertas de forma opcional cuando se produzcan determinados eventos. De forma predeterminada, ElastiCache registra los eventos relacionados con sus recursos, como la conmutación por error, el reemplazo de nodos, la operación de escalado y el mantenimiento programado, entre otros. Cada evento incluye la fecha y la hora, el nombre y el tipo de origen y una descripción.

Ventaja a nivel de pregunta: La capacidad de comprender y gestionar los motivos subyacentes de los eventos que activan las alertas generadas por su clúster le permite operar de manera más eficaz y responder a los eventos de manera adecuada.

- [Obligatorio] Revise los eventos generados por ElastiCache en la consola de ElastiCache (después de seleccionar su región) o mediante el comando describe-events de la Interfaz de línea de comandos de Amazon (AWS CLI) y la API de ElastiCache. Configure ElastiCache para que envíe notificaciones de los eventos de clúster importantes mediante Amazon Simple Notification Service (Amazon SNS). El uso de Amazon SNS con sus clústeres le permite realizar acciones mediante programación en los eventos de ElastiCache.
  - Hay dos grandes categorías de eventos: eventos actuales y programados. La lista de eventos actuales incluye: creación y eliminación de recursos, operaciones de escalado, conmutación por error, reinicio de nodos, creación de instantáneas, modificación de los parámetros del clúster, renovación del certificado de CA, eventos de error (error de aprovisionamiento del clúster [VPC o ENI], errores de escalado [ENI] y errores de instantáneas). La lista de eventos programados incluye: reemplazo de nodos programado durante el período de mantenimiento y reemplazo de nodos reprogramado.

- Si bien es posible que no necesite reaccionar de inmediato ante algunos de estos eventos, es fundamental analizar primero todos los eventos de error:
  - ElastiCache:AddCacheNodeFailed
  - ElastiCache:CacheClusterProvisioningFailed
  - ElastiCache:CacheClusterScalingFailed
  - ElastiCache:CacheNodesRebooted
  - ElastiCache:SnapshotFailed (solo Valkey o Redis OSS)
- [Recursos]:
  - Administración de notificaciones de Amazon SNS de ElastiCache
  - Notificaciones de eventos y Amazon SNS
- [Lo mejor] Para automatizar las respuestas a los eventos, utilice las capacidades de los productos y servicios de AWS, como las funciones de Lambda y SNS. Siga las prácticas recomendadas y haga cambios pequeños, frecuentes y reversibles, como código para que sus operaciones evolucionen a lo largo del tiempo. Utilice las métricas de Amazon CloudWatch para supervisar los clústeres.

  [Recursos]: Monitor ElastiCache (cluster mode disabled) read replica endpoints using AWS Lambda, Amazon Route 53, and Amazon SNS para un caso de uso que emplea Lambda y SNS.

OE 2: ¿Cuándo y cómo se escalan los clústeres de ElastiCache existentes?

Introducción a nivel de pregunta: El dimensionamiento correcto del clúster de ElastiCache es un acto de equilibrio que debe evaluarse cada vez que se produzcan cambios en los tipos de carga de trabajo subyacentes. El objetivo es operar con el entorno del tamaño adecuado para la carga de trabajo.

Ventaja a nivel de pregunta: La sobreutilización de los recursos puede provocar una latencia elevada y una disminución general del rendimiento. Por otro lado, la infrautilización puede provocar un aprovisionamiento excesivo de recursos con una optimización de costos deficiente. El dimensionamiento correcto de los entornos, puede lograr un equilibrio entre la eficiencia del rendimiento y la optimización de costos. Para corregir la utilización excesiva o insuficiente de los recursos, ElastiCache puede escalarse en dos dimensiones. Puede escalar verticalmente aumentando o disminuyendo la capacidad de los nodos. Y puede escalar horizontalmente añadiendo y eliminando nodos.

- [Obligatorio] Resuelva la sobreutilización de la CPU y la red en los nodos principales: descargue y redirija las operaciones de lectura a los nodos de réplica. Utilice nodos de réplica para las operaciones de lectura a fin de reducir la utilización del nodo principal. Para configurar esto en la biblioteca cliente de Valkey o Redis OSS, conéctese al punto de conexión del lector de ElastiCache para el modo de clúster deshabilitado o utilice el comando READONLY para el modo de clúster habilitado.

  [Recursos]:

  - Búsqueda de puntos de conexión en ElastiCache
  - Cluster Right-Sizing
  - Comando READONLY
- [Obligatorio] Supervise la utilización de los recursos cruciales del clúster, como la CPU, la memoria y la red. Es necesario realizar un seguimiento del uso de estos recursos específicos del clúster para tomar una decisión de escalado y determinar el tipo de operación de escalado. Si el modo de clúster de ElastiCache está deshabilitado, los nodos principales y de réplica pueden escalar verticalmente. Los nodos de réplica también pueden escalar horizontalmente de 0 a 5 nodos. Si el modo de clúster está habilitado, se aplica esto mismo a cada partición del clúster. Además, puede aumentar o reducir el número de particiones.

  [Recursos]:

  - Monitoring best practices with ElastiCache using Amazon CloudWatch
  - Escalado de clústeres de ElastiCache para Valkey y Redis OSS
  - Escalado de clústeres de ElastiCache para Memcached
- [Lo mejor] Supervisar el rendimiento a lo largo del tiempo puede ayudar a detectar cambios en la carga de trabajo que pasarían desapercibidos si se hiciera en un momento determinado. Para detectar tendencias a más largo plazo, utilice las métricas de CloudWatch para analizar intervalos de tiempo más largos. Los conocimientos obtenidos al observar períodos prolongados de métricas de CloudWatch deberían servir de base para pronosticar el uso de los recursos del clúster. Los datos y las métricas de CloudWatch están disponibles durante un máximo de 455 días.

  [Recursos]:

  - Supervisión de ElastiCache con métricas de CloudWatch
  - Monitoring Memcached with CloudWatch Metrics
  - Monitoring best practices with ElastiCache using Amazon CloudWatch

- [Lo mejor] Si sus recursos de ElastiCache se crean con CloudFormation, se recomienda realizar los cambios mediante plantillas de CloudFormation para preservar la coherencia operativa y evitar cambios de configuración no administrados y desviaciones de pila.

  [Recursos]:

  - ElastiCache resource type reference for CloudFormation
- [Lo mejor] Automatice sus operaciones de escalado mediante los datos operativos del clúster y defina umbrales en CloudWatch para configurar las alarmas. Utilice CloudWatch Events y Simple Notification Service (SNS) para activar funciones de Lambda y ejecutar una API de ElastiCache para escalar los clústeres automáticamente. Un ejemplo sería añadir una partición al clúster cuando la métrica `EngineCPUUtilization` alcance el 80 % durante un período de tiempo prolongado. Otra opción sería utilizar `DatabaseMemoryUsedPercentages` como umbral basado en la memoria.

  [Recursos]:

  - Using Amazon CloudWatch Alarms
  - ¿Qué es Amazon CloudWatch Events?
  - Uso de AWS Lambda con Amazon Simple Notification Service
  - Referencia de la API de ElastiCache

OE 3: ¿Cómo se administran los recursos del clúster de ElastiCache y se mantiene actualizado?

Introducción a nivel de pregunta: Al operar a escala, es fundamental que pueda localizar e identificar todos sus recursos de ElastiCache. A la hora de implementar nuevas características de la aplicación, debe crear una simetría entre las versiones del clúster en todos los tipos de entornos de ElastiCache: desarrollo, pruebas y producción. Los atributos de los recursos permiten separar los entornos para diferentes objetivos operativos, por ejemplo, cuando se implementan nuevas características y habilitan nuevos mecanismos de seguridad.

Ventaja a nivel de pregunta: La separación de los entornos de desarrollo, pruebas y producción es la mejor práctica operativa. También se recomienda aplicar los parches de software más recientes a los clústeres y nodos de todos los entornos mediante procesos bien conocidos y documentados. Al aprovechar las características nativas de ElastiCache, su equipo de ingeniería puede centrarse en cumplir los objetivos empresariales y no en el mantenimiento de ElastiCache.

- [Lo mejor] Ejecute el motor con la versión más reciente disponible y aplique las actualizaciones de autoservicio tan pronto como estén disponibles. ElastiCache actualiza automáticamente su infraestructura subyacente durante el período de mantenimiento especificado del clúster. Sin embargo, los nodos que se ejecutan en sus clústeres se actualizan mediante actualizaciones de autoservicio. Estas actualizaciones pueden ser de dos tipos: parches de seguridad o actualizaciones de software menores. Asegúrese de entender la diferencia entre los tipos de parches y cuándo se aplican.

  [Recursos]:

  - Actualizaciones de autoservicio en Amazon ElastiCache
  - Página de ayuda de actualizaciones de servicio y mantenimiento administrado de Amazon ElastiCache
- [Lo mejor] Organice y administre sus recursos de ElastiCache mediante etiquetas. Utilice las etiquetas en los grupos de replicación y no en los nodos individuales. Puede configurar las etiquetas de modo que se muestren cuando consulte los recursos y puede utilizarlas para realizar búsquedas y aplicar filtros. Debe utilizar los grupos de recursos para crear y mantener fácilmente colecciones de recursos que compartan conjuntos comunes de etiquetas.

  [Recursos]:

  - Tagging Best Practices
  - ElastiCache resource type reference for CloudFormation
  - Grupos de parámetros

OE 4: ¿Cómo se administran las conexiones de los clientes a sus clústeres de ElastiCache?

Introducción a nivel de pregunta: Al operar a escala, debe comprender cómo se conectan sus clientes con el clúster de ElastiCache para administras los aspectos operativos de la aplicación (como los tiempos de respuesta).

Ventaja a nivel de pregunta: Al elegir el mecanismo de conexión más adecuado, se garantiza que la aplicación no se desconecte debido a errores de conectividad, como los tiempos de espera.

- [Obligatorio] Separe las operaciones de lectura de las de escritura y conéctese a los nodos de réplica para ejecutar las operaciones de lectura. Sin embargo, tenga en cuenta que al separar las operaciones de escritura de las de lectura, perderá la capacidad de leer una clave inmediatamente después de escribirla debido a la naturaleza asincrónica de la replicación de Valkey y Redis OSS.

Se puede utilizar el comando WAIT para mejorar la seguridad de los datos en el mundo real y obligar a las réplicas a confirmar las escrituras antes de responder a los clientes, con un costo de rendimiento general. El uso de nodos de réplica para las operaciones de lectura se puede configurar en la biblioteca cliente de ElastiCache mediante el punto de conexión de lector de ElastiCache para el modo de clúster deshabilitado. Para el modo de clúster habilitado, utilice el comando READONLY. En muchas de las bibliotecas de cliente de ElastiCache, READONLY se implementa de forma predeterminada o mediante la configuración.

[Recursos]:

- Búsqueda de puntos de conexión en ElastiCache
- READONLY

- [Obligatorio] Utilice la agrupación de conexiones. El establecimiento de una conexión TCP tiene un costo en términos de tiempo de CPU tanto para el cliente como para el servidor, y la agrupación permite reutilizar la conexión TCP.

  Para reducir la sobrecarga de conexiones, debe utilizar la agrupación de conexiones. Con un conjunto de conexiones, la aplicación puede reutilizar y liberar conexiones “a voluntad”, sin el costo de establecer la conexión. Para implementar la agrupación de conexiones mediante la biblioteca cliente de ElastiCache (si es compatible) con un marco disponible para su entorno de aplicaciones o créelo desde cero.
- [Lo mejor] Asegúrese de que el tiempo de espera del socket del cliente esté establecido en, al menos, un segundo (en lugar del típico valor predeterminado de “ninguno” en numerosos clientes).
  - Si se establece un valor de tiempo de espera demasiado bajo, es posible que se produzcan tiempos de espera cuando la carga del servidor es alta. Si se establece demasiado alto, la aplicación puede tardar mucho en detectar problemas de conexión.
  - Controle el volumen de nuevas conexiones a través de la implementación de la agrupación de conexiones en su aplicación cliente. Esto reduce la latencia y el uso de CPU necesarios para abrir y cerrar las conexiones y realizar un protocolo de enlace de TLS si TLS está habilitado en el clúster.

  [Recursos]: Configure ElastiCache for higher availability
- [Bueno] El uso de la canalización (cuando los casos de uso lo permitan) puede aumentar significativamente el rendimiento.
  - La canalización reduce el tiempo de ida y vuelta (RTT) entre los clientes de la aplicación y el clúster, y es posible procesar nuevas solicitudes incluso si el cliente aún no ha leído las respuestas anteriores.

- Con la canalización, puede enviar varios comandos al servidor sin esperar a recibir respuestas o confirmaciones. La desventaja de la canalización es que, cuando finalmente se obtienen todas las respuestas de forma masiva, es posible que se haya producido un error que no se detectará hasta el final.
- Implemente métodos para reintentar las solicitudes cuando se devuelva un error que omita la solicitud incorrecta.

[Recursos]: Pipelining

OE 5: ¿Cómo se implementan los componentes de ElastiCache para una carga de trabajo?

Introducción a nivel de pregunta: Los entornos de ElastiCache se pueden implementar manualmente a través de la consola de AWS o mediante programación mediante API, CLI, kits de herramientas, etc. Las prácticas recomendadas de excelencia operativa sugieren automatizar las implementaciones mediante código siempre que sea posible. Además, los clústeres de ElastiCache pueden aislarse por carga de trabajo o combinarse para optimizar los costos.

Ventaja a nivel de pregunta: La elección del mecanismo de implementación más adecuado para sus entornos de ElastiCache puede mejorar la excelencia operativa con el tiempo. Se recomienda realizar operaciones como código siempre que sea posible para minimizar los errores humanos y aumentar la repetibilidad, la flexibilidad y el tiempo de respuesta a los eventos.

Al comprender los requisitos de aislamiento de la carga de trabajo, puede optar por tener entornos de ElastiCache dedicados por carga de trabajo o combinar varias cargas de trabajo en clústeres individuales o combinaciones de los mismos. Comprender las soluciones de compromiso puede ayudar a lograr un equilibrio entre la excelencia operativa y la optimización de costos.

- [Obligatorio] Comprenda las opciones de implementación disponibles para ElastiCache y automatice estos procedimientos siempre que sea posible. Las posibles vías de automatización incluyen CloudFormation, AWS CLI/SDK y API.

  [Recursos]:

  - Amazon ElastiCache resource type reference
  - elasticache
  - Referencia de la API de Amazon ElastiCache

- [Obligatorio] Para todas las cargas de trabajo, determine el nivel de aislamiento del clúster necesario.
  - [Lo mejor]: Alto aislamiento: asignación 1:1 de la carga de trabajo al clúster. Permite un control más preciso sobre el acceso, el tamaño, el escalado y la administración de los recursos de ElastiCache por carga de trabajo.
  - [Mejor] Aislamiento medio: M:1 aislado por propósito, pero quizás compartido entre varias cargas de trabajo (por ejemplo, un clúster dedicado al almacenamiento en caché de las cargas de trabajo y otro dedicado a la mensajería).
  - [Bueno]: Bajo aislamiento: M:1 multiusos, totalmente compartido. Recomendado para cargas de trabajo en las que es aceptable el acceso compartido.

OE 6: ¿Cómo se planifican y mitigan los errores?

Introducción a nivel de pregunta: La excelencia operativa incluye anticipar los errores mediante la realización de ejercicios “ante mortem” periódicos para identificar las posibles fuentes de errores y poder eliminarlas o mitigarlas. ElastiCache ofrece una API de conmutación por error que permite simular eventos de error en los nodos con fines de prueba.

Ventaja a nivel de pregunta: Al probar los escenarios de error con antelación, puede averiguar cómo afectan a su carga de trabajo. Esto permite probar de forma segura los procedimientos de respuesta y su eficacia. Además su equipo se familiarizará con su ejecución.

[Obligatorio] Realice pruebas de conmutación por error periódicas en las cuentas de desarrollo/pruebas. TestFailover

OE 7: ¿Cómo se solucionan los problemas de los eventos del motor de Valkey o Redis OSS?

Introducción a nivel de pregunta: La excelencia operativa requiere la capacidad de investigar tanto la información a nivel de servicio como a nivel de motor para analizar la condición y el estado de sus clústeres. ElastiCache puede emitir registros del motor de Valkey o Redis OSS tanto a Amazon CloudWatch como a Amazon Kinesis Data Firehose.

Ventaja relacionada con preguntas: al habilitar los registros del motor de Valkey o Redis OSS en los clústeres de Amazon ElastiCache, se proporciona información sobre los eventos que afectan al estado y al rendimiento de los clústeres. Los registros del motor de Valkey o Redis OSS proporcionan datos directamente del motor que no están disponibles a través del mecanismo de

eventos de ElastiCache. Mediante una observación detenida de los eventos de ElastiCache (consulte OE-1 más arriba) y de los registros del motor, es posible determinar el orden de los eventos a la hora de solucionar problemas tanto desde la perspectiva del servicio de ElastiCache como desde la perspectiva del motor.

- [Obligatorio] Asegúrese de que la funcionalidad de registro del motor de Redis OSS esté habilitada. Esta funcionalidad está disponible en ElastiCache versión 6.2 para Redis OSS y posteriores. Esto se puede hacer durante la creación del clúster o modificándolo después de la creación.
  - Determine si los Registros de Amazon CloudWatch o Amazon Kinesis Data Firehose son el destino adecuado para los registros del motor de Redis OSS.
  - Seleccione un registro de destino adecuado en CloudWatch o Kinesis Data Firehose para conservar los registros. Si tiene varios clústeres, considere la posibilidad de utilizar un registro de destino diferente para cada clúster, ya que esto ayudará a aislar los datos a la hora de solucionar problemas.

  [Recursos]:

  - Entrega de registros: Entrega de registro
  - Destinos de registro: Registros de Amazon CloudWatch
  - Introducción a los Registros de Amazon CloudWatch ¿Qué es Amazon CloudWatch Logs?
  - Introducción a Amazon Kinesis Data Firehose: ¿Qué es Amazon Kinesis Data Firehose?
- [Lo mejor] Si utiliza Registros de Amazon CloudWatch, considere la posibilidad de utilizar la Información de registros de Amazon CloudWatch para consultar el registro del motor de Valkey o Redis OSS y obtener información importante.

  Por ejemplo, cree una consulta en el grupo de registros de CloudWatch que contenga los registros del motor de Valkey o Redis OSS y devuelva los eventos con un nivel de registro de ADVERTENCIA, como:

  ```
  fields @timestamp, LogLevel, Message
  | sort @timestamp desc
  | filter LogLevel = "WARNING"
  ```

  [Recursos]:Análisis de los datos de registros con Amazon CloudWatch Logs Insights

Pilar de seguridad del enfoque Well-Architected de Amazon ElastiCache

El pilar de seguridad se centra en proteger la información y los sistemas. Los temas clave incluyen la confidencialidad e integridad de los datos, la identificación y administración de quién puede hacer qué con la administración de privilegios, la protección de los sistemas y el establecimiento de controles para detectar eventos de seguridad.

Temas

- SEC 1: ¿Qué medidas está tomando para controlar el acceso autorizado a los datos de ElastiCache?
- SEC 2: ¿Requieren sus aplicaciones autorización adicional para ElastiCache además de los controles basados en red?
- SEC 3: ¿Existe el riesgo de que los comandos se ejecuten de forma inadvertida y provoquen la pérdida de datos o errores?
- SEC 4: ¿Cómo se garantiza el cifrado de datos en reposo con ElastiCache?
- SEC 5: ¿Cómo se cifran los datos en tránsito con ElastiCache?
- SEC 6: ¿Cómo se restringe el acceso a los recursos del plano de control?
- SEC 7: ¿Cómo se detectan los eventos de seguridad y cómo se responde a ellos?

SEC 1: ¿Qué medidas está tomando para controlar el acceso autorizado a los datos de ElastiCache?

Introducción a nivel de pregunta: Todos los clústeres de ElastiCache están diseñados de forma que es posible acceder a ellos desde instancias de Amazon Elastic Compute Cloud en una VPC, funciones sin servidor (AWS Lambda) o contenedores (Amazon Elastic Container Service). El escenario más frecuente es obtener acceso a un clúster de ElastiCache desde una instancia de Amazon Elastic Compute Cloud en la misma Amazon Virtual Private Cloud (Amazon Virtual Private Cloud). Antes de poder conectarse a un clúster desde una instancia de Amazon EC2, debe autorizar a la instancia de Amazon EC2 el acceso al clúster. Para acceder a un clúster de ElastiCache que se ejecuta en una VPC, es necesario conceder acceso de red al clúster.

Ventaja a nivel de pregunta: El acceso a la red en el clúster se controla mediante grupos de seguridad de VPC. Un grupo de seguridad funciona como un firewall virtual para las instancias de Amazon EC2 para controlar el tráfico entrante y saliente. Las reglas de entrada controlan el tráfico entrante a la instancia y las reglas de salida controlan el tráfico saliente desde la instancia. En el caso de ElastiCache, al lanzar un clúster, es necesario asociar un grupo de seguridad. Esto garantiza

que haya reglas de tráfico entrante y saliente vigentes para todos los nodos que componen el clúster. Además, ElastiCache está configurado para desplegarse exclusivamente en subredes privadas, de modo que solo se pueda acceder a ellas a través de la red privada de la VPC.

- [Obligatorio] El grupo de seguridad asociado al clúster controla el acceso a la red y al clúster. De forma predeterminada, los grupos de seguridad no tienen ninguna regla de acceso definida y, por lo tanto, no habrá ninguna ruta de acceso a ElastiCache. Para habilitar esto, configure una regla de entrada en el grupo de seguridad que especifique la dirección/rango de IP de origen, el tráfico de tipo TCP y el puerto del clúster de ElastiCache (el puerto predeterminado 6379 para ElastiCache para Valkey y Redis OSS, por ejemplo). Si bien es posible permitir un conjunto muy amplio de orígenes de acceso, como todos los recursos de una VPC (0.0.0.0/0), se recomienda ser lo más detallado posible al definir las reglas de acceso, por ejemplo, autorizar solo el acceso entrante a los clientes de Valkey o Redis OSS que se ejecuten en instancias de Amazon EC2 asociadas a un grupo de seguridad concreto.

  [Recursos]:

  - Subredes y grupos de subredes
  - Acceso al clúster o al grupo de reproducción
  - Controlar el tráfico hacia los recursos mediante grupos de seguridad
  - Grupos de seguridad de Amazon Elastic Compute Cloud para instancias de Linux
- [Obligatorio] Se pueden asignar políticas de AWS Identity and Access Management a funciones de AWS Lambda para que puedan acceder a los datos de ElastiCache. Para habilitar esta característica, cree un rol de ejecución de IAM con el permiso `AWSLambdaVPCAccessExecutionRole` y, a continuación, asigne el rol a la función de AWS Lambda.

  [Recursos]: Configuración de una función de Lambda para acceder a Amazon ElastiCache en una Amazon VPC: Tutorial: Configuración de una función de Lambda para obtener acceso a Amazon ElastiCache en una Amazon VPC

SEC 2: ¿Requieren sus aplicaciones autorización adicional para ElastiCache además de los controles basados en red?

Introducción en la pregunta: En los escenarios en los que sea necesario restringir o controlar el acceso a los clústeres en el cliente individual, se recomienda realizar la autenticación con el comando AUTH. Los tokens de autenticación de ElastiCache, con administración opcional de

usuarios y grupos de usuarios, permiten que ElastiCache requiera una contraseña antes de permitir a los clientes ejecutar comandos y claves de acceso, lo que mejora la seguridad del plano de datos.

Ventaja en la pregunta: A fin de ayudar a mantener la seguridad de los datos, ElastiCache proporciona mecanismos de protección contra el acceso no autorizado a los datos. Entre ellos se incluye obligar a los clientes a utilizar AUTH, o token AUTH (contraseña), de control de acceso basado en roles (RBAC) para conectarse a ElastiCache antes de ejecutar los comandos autorizados.

- [Lo mejor] Para ElastiCache versión 6.x y posteriores para Redis OSS, y ElastiCache versión 7.2 y posteriores para Valkey, defina los controles de autenticación y autorización con grupos de usuarios, usuarios y cadenas de acceso. Asigne usuarios a grupos de usuarios y, a continuación, asigne grupos de usuarios a clústeres. Para utilizar el RBAC, este debe seleccionarse al crear el clúster y debe estar habilitado el cifrado en tránsito. Asegúrese de utilizar un cliente de Valkey o Redis que sea compatible con TLS para poder utilizar RBAC.

  [Recursos]:

  - Aplicación de RBAC a un grupo de replicación para ElastiCache
  - Especificación de permisos mediante una cadena de acceso
  - ACL de
  - Versiones de ElastiCache compatibles
- [Lo mejor] Para las versiones de ElastiCache anteriores a la 6.x para Redis OSS, además de establecer un token o contraseña seguros y mantener una política de contraseñas estricta para AUTH, se recomienda rotar la contraseña o token. ElastiCache puede administrar hasta dos (2) tokens de autenticación en un momento dado. También puede modificar el clúster para que requiera explícitamente el uso de tokens de autenticación.

  [Recursos]: Modificación del token AUTH en un clúster de ElastiCache existente

SEC 3: ¿Existe el riesgo de que los comandos se ejecuten de forma inadvertida y provoquen la pérdida de datos o errores?

Introducción de pregunta: hay varios comandos de Valkey o Redis OSS que pueden tener una repercusión negativa en las operaciones si se ejecutan por error o por parte de personas malintencionadas. Estos comandos pueden tener consecuencias no deseadas desde el punto de vista del rendimiento y la seguridad de los datos. Por ejemplo, un desarrollador puede llamar de forma rutinaria al comando FLUSHALL en un entorno de desarrollo y, debido a un error, puede

intentar ejecutar este comando sin darse cuenta en un sistema de producción, lo que provoca la pérdida accidental de datos.

Ventaja en la pregunta: A partir de ElastiCache versión 5.0.3 para Redis OSS, se puede cambiar el nombre de algunos comandos que podrían perjudicar a la carga de trabajo. El cambio del nombre de los comandos puede ayudar a evitar que se ejecuten inadvertidamente en el clúster.

- [Obligatorio]

  [Recursos]:

  - ElastiCache versión 5.0.3 para Redis OSS (obsoleta, use la versión 5.0.6)
  - Cambios en los parámetros de ElastiCache versión 5.0.3 para Redis OSS
  - Redis OSS security

SEC 4: ¿Cómo se garantiza el cifrado de datos en reposo con ElastiCache?

Introducción en la pregunta: Si bien ElastiCache es un almacén de datos en memoria, se puede cifrar cualquier dato que pueda persistir (en el almacenamiento) como parte de las operaciones estándar del clúster. Esto incluye las copias de seguridad programadas y manuales escritas en Amazon S3, así como los datos guardados en el almacenamiento en disco como resultado de las operaciones de sincronización e intercambio. Los tipos de instancias de las familias M6g y R6g también cuentan con un cifrado en memoria siempre activo.

Ventaja en la pregunta: ElastiCache proporciona un cifrado opcional en reposo para aumentar la seguridad de los datos.

- [Obligatorio] El cifrado en reposo solo se puede habilitar en un clúster de ElastiCache (grupo de replicación) en el momento de su creación. No se puede modificar un clúster existente para empezar a cifrar los datos en reposo. De forma predeterminada, ElastiCache proporcionará y administrará las claves utilizadas en el cifrado en reposo.

  [Recursos]:

  - Restricciones del cifrado en reposo
  - Activación del cifrado en reposo
- [Lo mejor] Aproveche los tipos de instancias de Amazon EC2 que cifran los datos mientras están en la memoria (como M6g o R6g). Siempre que sea posible, considere la posibilidad de administrar sus propias claves para el cifrado en reposo. Para entornos de seguridad de datos más estrictos, se puede utilizar AWS Key Management Service (KMS) para autogestionar las claves maestras de

cliente (CMK). Mediante la integración de ElastiCache con AWS Key Management Service, puede crear, poseer y administrar las claves utilizadas para el cifrado de datos en reposo de su clúster de ElastiCache.

[Recursos]:

- Uso de claves administradas por el cliente desde AWS Key Management Service
- AWS Key Management Service
- AWS KMS concepts

SEC 5: ¿Cómo se cifran los datos en tránsito con ElastiCache?

Introducción a nivel de pregunta: Es un requisito habitual evitar que los datos corran peligro mientras están en tránsito. Esto representa los datos dentro de los componentes de un sistema distribuido, así como entre los clientes de la aplicación y los nodos del clúster. ElastiCache es compatible con este requisito pues permite cifrar los datos en tránsito entre los clientes y el clúster, y entre los propios nodos del clúster. Los tipos de instancias de las familias M6g y R6g también cuentan con un cifrado en memoria siempre activo.

Ventaja a nivel de pregunta: El cifrado en tránsito de Amazon ElastiCache es una característica opcional que permite reforzar la seguridad de los datos en sus momentos más vulnerables: cuando se trasladan de una ubicación a otra.

- [Obligatorio] El cifrado en tránsito solo se puede habilitar en un clúster (grupo de replicación) en el momento de su creación. Tenga en cuenta que, debido al procesamiento adicional necesario para cifrar o descifrar datos, la implementación del cifrado en tránsito afectará al rendimiento en cierta medida. Para entender el impacto, se recomienda comparar la carga de trabajo antes y después de habilitar el cifrado en tránsito.

  [Recursos]:

  - Información general sobre el cifrado en tránsito

SEC 6: ¿Cómo se restringe el acceso a los recursos del plano de control?

Introducción en la pregunta: Las políticas de IAM y el ARN facilitan los controles de acceso detallados para ElastiCache para Valkey y Redis OSS, lo que permite llevar a cabo un control más estricto para administrar la creación, modificación y eliminación de los clústeres.

Ventaja a nivel de pregunta: La administración de los recursos de Amazon ElastiCache, como los grupos de replicación, los nodos, etc., puede restringirse a las cuentas de AWS que tengan permisos específicos según las políticas de IAM, lo que mejora la seguridad y la fiabilidad de los recursos.

- [Obligatorio] Gestione el acceso a los recursos de Amazon ElastiCache mediante la asignación de políticas de AWS Identity and Access Management específicas a los usuarios de AWS, lo que permite un control más preciso sobre qué cuentas pueden realizar qué acciones en los clústeres.

  [Recursos]:

  - Información general sobre la administración de los permisos de acceso a los recursos de ElastiCache
  - Uso de políticas basadas en identidad (políticas de IAM) para Amazon ElastiCache

SEC 7: ¿Cómo se detectan los eventos de seguridad y cómo se responde a ellos?

Introducción a nivel de pregunta: ElastiCache, cuando se implementa con el RBAC habilitado, exporta las métricas de CloudWatch para notificar a los usuarios los eventos de seguridad. Estas métricas ayudan a identificar los intentos fallidos de autenticación, acceso a claves o ejecución de comandos para los que los usuarios con RBAC que se conectan no están autorizados.

Además, los recursos de productos y servicios de AWS ayudan a proteger la carga de trabajo general al automatizar las implementaciones y registrar todas las acciones y modificaciones para su posterior revisión o auditoría.

Ventaja a nivel de pregunta: Con la supervisión de los eventos, usted permite a su organización responder de acuerdo con sus requisitos, políticas y procedimientos. La automatización de la supervisión y las respuestas a estos eventos de seguridad refuerza su postura de seguridad general.

- [Obligatorio] Familiarícese con las métricas de CloudWatch publicadas relacionadas con los errores de autenticación y autorización del RBAC.
  - AuthenticationFailures = Intentos fallidos de autenticación en Valkey o Redis OSS
  - KeyAuthorizationFailures = Intentos fallidos de los usuarios de acceder a las claves sin permiso
  - CommandAuthorizationFailures = Intentos fallidos de los usuarios de ejecutar comandos sin permiso

  [Recursos]:

  - Métricas de Valkey y Redis OSS

- [Lo mejor] Se recomienda configurar alertas y notificaciones para estas métricas y responder según sea necesario.

  [Recursos]:

  - Uso de las alarmas de Amazon CloudWatch
- [Lo mejor] Utilice el comando de la ACL LOG de Valkey o Redis OSS para recopilar más detalles.

  [Recursos]:

  - ACL LOG
- [Lo mejor] Familiarícese con las capacidades de los productos y servicios de AWS en lo que respecta a la supervisión, el registro y el análisis de las implementaciones y eventos de ElastiCache.

  [Recursos]:

  - Registro de llamadas a la API de Amazon ElastiCache con AWS
  - elasticache-redis-cluster-automatic-backup-check
  - Monitoreo del uso con métricas de CloudWatch

Pilar de fiabilidad del enfoque Well-Architected de Amazon ElastiCache

El pilar de la fiabilidad se centra en las cargas de trabajo que realizan las funciones previstas y en cómo recuperarse rápidamente de la incapacidad de satisfacer las demandas. Entre los temas clave, se incluyen el diseño de sistemas distribuidos, la planificación de la recuperación y cómo adaptarse a los cambios de requisitos.

Temas

- REL 1: ¿Cómo se respaldan las implementaciones de arquitecturas de alta disponibilidad (HA)?
- REL 2: ¿Cómo se cumplen los objetivos de puntos de recuperación (RPO) con ElastiCache?
- REL 3: ¿Cómo se cumplen los requisitos de recuperación de desastres (DR)?
- REL 4: ¿Cómo se planifican eficazmente las conmutaciones por error?
- REL 5: ¿Están sus componentes de ElastiCache diseñados para el escalado?

REL 1: ¿Cómo se respaldan las implementaciones de arquitecturas de alta disponibilidad (HA)?

Introducción a nivel de pregunta: Comprender la arquitectura de alta disponibilidad de Amazon ElastiCache le permitirá operar en un estado resiliente durante los eventos de disponibilidad.

Ventaja a nivel de pregunta: Diseñar los clústeres de ElastiCache para que sean resilientes a los errores garantiza una mayor disponibilidad para las implementaciones de ElastiCache.

- [Obligatorio] Determine el grado de fiabilidad que necesita para su clúster de ElastiCache. Las diferentes cargas de trabajo tienen diferentes estándares de resiliencia, desde cargas de trabajo totalmente efímeras hasta cargas de trabajo cruciales para la misión. Defina las necesidades de cada tipo de entorno en el que opere, como desarrollo, pruebas y producción.

  Motor de almacenamiento en caché: ElastiCache para Memcached frente a ElastiCache para Valkey y Redis OSS

  1. ElastiCache para Memcached no proporciona ningún mecanismo de replicación y se usa principalmente para cargas de trabajo efímeras.
  2. ElastiCache para Valkey y Redis OSS ofrece las características de HA que se describen a continuación
- [Lo mejor] Para las cargas de trabajo que requieren HA, utilice ElastiCache en modo clúster con un mínimo de dos réplicas por partición, incluso para cargas de trabajo pequeñas con requisitos de rendimiento de una sola partición.
  1. Si el modo de clúster está activado, multi-AZ se habilita automáticamente.

     Multi-AZ minimiza el tiempo de inactividad, ya que realiza conmutaciones por error automáticas desde el nodo principal a las réplicas, en caso de mantenimiento planificado o no planificado, y mitiga los errores de AZ.
  2. En cuanto a las cargas de trabajo particionadas, un mínimo de tres particiones proporciona una recuperación más rápida durante los eventos de conmutación por error, ya que el protocolo de clúster de Valkey o Redis OSS requiere que la mayoría de los nodos principales estén disponibles para alcanzar el quórum.
  3. Configure dos o más réplicas para ofrecer disponibilidad.

     Tener dos réplicas proporciona escalabilidad de lectura mejorada así como disponibilidad de lectura en situaciones en las que una réplica está en mantenimiento.

4. Utilice tipos de nodos basados en Graviton2 (nodos predeterminados en la mayoría de las regiones).

   ElastiCache ha añadido rendimiento optimizado en estos nodos. Como resultado, obtiene un mejor rendimiento de replicación y sincronización, lo que se traduce en una mejora general de la disponibilidad.
5. Supervise y ajuste el tamaño para hacer frente a los picos de tráfico previstos: con una carga pesada, el motor puede dejar de responder, lo que afecta a la disponibilidad. `BytesUsedForCache` y `DatabaseMemoryUsagePercentage` son buenos indicadores del uso de la memoria, mientras que `ReplicationLag` es un indicador del estado de la replicación en función de la velocidad de escritura. Puede usar estas métricas para activar el escalado de clústeres.
6. Para garantizar la resiliencia del cliente realice pruebas con la API de conmutación por error antes de un evento de conmutación por error en producción.

[Recursos]:

- Configure ElastiCache for Redis OSS for higher availability
- Alta disponibilidad a través de grupos de reproducción

REL 2: ¿Cómo se cumplen los objetivos de puntos de recuperación (RPO) con ElastiCache?

Introducción a nivel de pregunta: Comprenda el RPO de la carga de trabajo para tomar decisiones informadas sobre las estrategias de respaldo y recuperación de ElastiCache.

Ventaja a nivel de pregunta: Tener una estrategia de RPO implementada le permite mejorar la continuidad del negocio en caso de que se produzcan escenarios de recuperación de desastres. El diseño de políticas de respaldo y restauración puede contribuir a cumplir los objetivos de puntos de recuperación (RPO) de sus datos de ElastiCache. ElastiCache ofrece capacidades de instantáneas que se almacenan en Amazon S3, junto con una política de retención configurable. Estas instantáneas se toman durante un período de copia de seguridad definido y el servicio las gestiona automáticamente. Si la carga de trabajo requiere un grado de detalle de respaldo adicional, tiene la opción de crear hasta 20 copias de seguridad manuales por día. Las copias de seguridad creadas manualmente no tienen una política de retención de servicios y se pueden conservar indefinidamente.

- [Obligatorio] Comprenda y documente el RPO de sus implementaciones de ElastiCache.

  - Tenga en cuenta que Memcached no ofrece ningún proceso de copia de seguridad.
  - Revise las capacidades de las características de copia de seguridad y restauración de ElastiCache.
- [Lo mejor]Disponga de un proceso que se haya comunicado bien para hacer copias de seguridad de su clúster.
  - Inicie copias de seguridad manuales según sea necesario.
  - Revise las políticas de retención de las copias de seguridad automáticas.
  - Tenga en cuenta que las copias de seguridad manuales se conservarán indefinidamente.
  - Programe las copias de seguridad automáticas durante los períodos de bajo uso.
  - Realice operaciones de respaldo contra réplicas de lectura para asegurarse de minimizar el impacto en el rendimiento del clúster.
- [Bueno] Aproveche la característica de copia de seguridad programada de ElastiCache para realizar copias de seguridad periódicas de los datos durante un período definido.
  - Haga pruebas periódicas de las restauraciones a partir de las copias de seguridad.
- [Recursos]:
  - Redis OSS
  - Copia de seguridad y restauración para ElastiCache
  - Realización de copias de seguridad manuales
  - Programación de copias de seguridad automáticas
  - Backup and Restore ElastiCache Clusters

REL 3: ¿Cómo se cumplen los requisitos de recuperación de desastres (DR)?

Introducción a nivel de pregunta: La recuperación de desastres es un aspecto importante de cualquier planificación de cargas de trabajo. ElastiCache ofrece varias opciones para implementar la recuperación ante desastres en función de los requisitos de resiliencia de la carga de trabajo. Con el almacén de datos global de Amazon ElastiCache puede escribir en su clúster en una región y disponer de los datos para leerlos en otros dos clústeres de réplica entre regiones, lo que permite lecturas de baja latencia y recuperación ante desastres en todas las regiones.

Ventaja a nivel de pregunta: Comprender y planificar una variedad de escenarios de desastres contribuye a garantizar la continuidad del negocio. Las estrategias de DR deben equilibrarse en relación al costo, el impacto en el rendimiento y la posible pérdida de datos.

- [Obligatorio] Desarrolle y documente estrategias de DR para todos los componentes de ElastiCache en función de los requisitos de la carga de trabajo. ElastiCache es único porque algunos casos de uso son completamente efímeros y no requieren ninguna estrategia de DR, mientras que otros se encuentran en el extremo opuesto del abanico y exigen una estrategia de DR extremadamente sólida. Todas las opciones deben soparse teniendo en cuenta la optimización de costos: una mayor resiliencia requiere más infraestructura.

  Conozca las opciones de DR disponibles a nivel regional y multirregional.

  - Se recomiendan implementaciones multi-AZ para evitar errores de AZ. Asegúrese de realizar la implementación con el modo de clúster habilitado en las arquitecturas multi-AZ, con un mínimo de tres AZ disponibles.
  - Se recomienda utilizar Global Datastore para protegerse contra los errores regionales.
- [Lo mejor] Habilite Global Datastore para cargas de trabajo que requieran resiliencia a nivel de región.
  - Tenga un plan para realizar una conmutación por error a la región secundaria en caso de degradación de la primaria.
  - Pruebe el proceso de conmutación por error multirregional antes de llevar a cabo una conmutación por error en producción.
  - Supervise la métrica `ReplicationLag` para conocer el posible impacto de la pérdida de datos durante los eventos de conmutación por error.
- [Recursos]:
  - Mitigación de errores
  - Replicación en regiones de AWS mediante los almacenes de datos globales
  - Restauración desde una copia de seguridad con redimensionamiento opcional del clúster
  - Minimización del tiempo de inactividad en ElastiCache para Valkey y Redis OSS con Multi-AZ

REL 4: ¿Cómo se planifican eficazmente las conmutaciones por error?

Introducción a nivel de pregunta: La habilitación multi-AZ con conmutaciones por error automáticas es una práctica recomendada de ElastiCache. En algunos casos, ElastiCache para Valkey y Redis OSS reemplaza los nodos principales como parte de las operaciones de servicio. Algunos ejemplos son eventos de mantenimiento planificado y el improbable caso de que haya un problema en el nodo o en la zona de disponibilidad. Las conmutaciones por error correctas dependen tanto de la configuración de ElastiCache como de la biblioteca cliente.

Ventaja en la pregunta: Al seguir las prácticas recomendadas para las conmutaciones por error de ElastiCache junto con la biblioteca cliente específica de ElastiCache se ayuda a minimizar el posible tiempo de inactividad durante los eventos de conmutación por error.

- [Obligatorio] Si el modo de clúster está deshabilitado, utilice los tiempos de espera para que sus clientes detecten si es necesario desconectarse del nodo principal anterior y volver a conectarse al nuevo nodo principal, mediante la dirección IP del punto de conexión principal actualizada. Si el modo de clúster está habilitado, la biblioteca cliente es responsable de detectar los cambios en la topología del clúster subyacente. La mayoría de las veces, esto se logra mediante ajustes de configuración de la biblioteca cliente de ElastiCache, que también permiten configurar la frecuencia y el método de actualización. Cada biblioteca cliente ofrece su propia configuración. Encontrará más información en la documentación correspondiente.

  [Recursos]:

  - Minimización del tiempo de inactividad en ElastiCache para Valkey y Redis OSS con Multi-AZ
  - Revise las prácticas recomendadas de su biblioteca cliente de ElastiCache.
- [Obligatorio] Las conmutaciones por error correctas dependen de un entorno de replicación en buen estado entre los nodos principal y de réplica. Revise y comprenda la naturaleza asincrónica de la replicación de Valkey y Redis OSS, así como las métricas de CloudWatch disponibles para informar sobre el retraso de replicación entre los nodos principales y de réplica. Para los casos de uso que requieran una mayor seguridad de los datos, utilice el comando WAIT para obligar a las réplicas a confirmar las escrituras antes de responder a los clientes conectados.

  [Recursos]:

  - Métricas de Valkey y Redis OSS
  - Monitoring best practices with ElastiCache using Amazon CloudWatch
- [Lo mejor] Valide periódicamente la capacidad de respuesta de su aplicación durante la conmutación por error mediante la API de conmutación por error de prueba de ElastiCache.

  [Recursos]:

  - Testing Automatic Failover to a Read Replica on ElastiCache
  - Prueba de la conmutación por error automática

REL 5: ¿Están sus componentes de ElastiCache diseñados para el escalado?

Introducción a nivel de pregunta: Al comprender las capacidades de escalado y las topologías de implementación disponibles, los componentes de ElastiCache pueden ajustarse con el tiempo para satisfacer los cambiantes requisitos de la carga de trabajo. ElastiCache ofrece escalado en 4 direcciones: reducir/escalar horizontalmente y reducir/escalar verticalmente.

Ventaja a nivel de pregunta: El seguimiento de las prácticas recomendadas para las implementaciones de ElastiCache proporciona la mayor flexibilidad de escalado, además de cumplir con el principio de Well-Architected de escalar horizontalmente para minimizar el impacto de los errores.

- [Obligatorio] Comprenda la diferencia entre las topologías de modo clúster habilitado y modo clúster deshabilitado. En casi todos los casos, se recomienda realizar la implementación con el modo de clúster habilitado, ya que permite una mayor escalabilidad a lo largo del tiempo. Los componentes en modo clúster deshabilitado tienen una capacidad limitada para escalar horizontalmente mediante la adición de réplicas de lectura.
- [Obligatorio] Comprenda cuándo y cómo escalar.
  - Para más READIOPS: añada réplicas
  - Para más WRITEOPS: añada particiones (escale horizontamente)
  - Para más E/S de red: utilice instancias optimizadas para la red y escale verticalmente
- [Lo mejor] Despliegue los componentes de ElastiCache con el modo de clúster habilitado, con una preferencia por más nodos más pequeños en lugar de menos nodos más grandes. Esto limita de manera efectiva el radio de acción de un error de nodo.
- [Lo mejor] Incluya réplicas en los clústeres para mejorar la capacidad de respuesta durante los eventos de escalado.
- [Bueno] Si el modo de clúster está deshabilitado, aproveche las réplicas de lectura para aumentar la capacidad de lectura general. ElastiCache ofrece compatibilidad con hasta 5 réplicas de lectura en modo de clúster deshabilitado, así como escalado vertical.
- [Recursos]:
  - Escalado de clústeres de ElastiCache
  - Escalado vertical en línea

Pilar de eficiencia del rendimiento del enfoque Well-Architected de Amazon ElastiCache

El pilar de eficiencia del rendimiento se centra en el uso eficiente de recursos informáticos y de TI. Los temas clave incluyen la selección de los tipos y tamaños de recursos correctos en función de los requisitos de la carga de trabajo, la supervisión del rendimiento y la toma de decisiones informadas para mantener la eficiencia a medida que evolucionan las necesidades empresariales.

Temas

- PE 1: ¿Cómo se supervisa el rendimiento del clúster de Amazon ElastiCache?
- PE 2: ¿Cómo se distribuye el trabajo entre los nodos del clúster de ElastiCache?
- PE 3: En el caso del almacenamiento en caché de las cargas de trabajo, ¿cómo se realiza el seguimiento de la eficacia y el rendimiento de la memoria caché y se informa al respecto?
- PE 4: ¿Cómo optimiza la carga de trabajo el uso de los recursos y las conexiones de red?
- PE 5: ¿Cómo se gestiona la eliminación o la expulsión de claves?
- PE 6: ¿Cómo se modelan los datos y se interactúa con ellos en ElastiCache?
- PE 7: ¿Cómo se registran los comandos de ejecución lenta en el clúster de Amazon ElastiCache?
- PE8: ¿Cómo ayuda el escalado automático a aumentar el rendimiento del clúster de ElastiCache?

PE 1: ¿Cómo se supervisa el rendimiento del clúster de Amazon ElastiCache?

Introducción a nivel de pregunta: Al comprender las métricas de supervisión existentes, puede identificar la utilización actual. La supervisión adecuada puede ayudar a identificar posibles obstáculos que afecten al rendimiento de su clúster.

Ventaja a nivel de pregunta: Comprender las métricas asociadas a su clúster puede ayudar a guiar las técnicas de optimización que pueden reducir la latencia y aumentar el rendimiento.

- [Obligatorio] Pruebas de rendimiento de referencia con un subconjunto de la carga de trabajo.
  - Debe supervisar el rendimiento de la carga de trabajo real mediante mecanismos como las pruebas de carga.
  - Supervise las métricas de CloudWatch mientras ejecuta estas pruebas para comprender las métricas disponibles y establecer una referencia de rendimiento.

- [Lo mejor] En el caso de las cargas de trabajo de ElastiCache para Valkey y Redis OSS, cambie el nombre de los comandos caros desde el punto de vista computacional, como `KEYS`, para limitar la capacidad de los usuarios de ejecutar comandos de bloqueo en los clústeres de producción.
  - Las cargas de trabajo de ElastiCache que ejecutan el motor 6.x para Redis OSS pueden aprovechar el control de acceso basado en roles para restringir determinados comandos. El acceso a los comandos se puede controlar con la creación de usuarios y grupos de usuarios con la consola o la CLI de AWS y con la asociación de los grupos de usuarios a un clúster. En Redis OSS 6, cuando RBAC está habilitado, es posible usar -@dangerous, lo que no permitirá comandos costosos, como KEYS, MONITOR, SORT, etc., para ese usuario.
  - En la versión 5.x del motor, cambie el nombre de los comandos mediante el parámetro `rename-commands` del grupo de parámetros del clúster.
- [Mejor] Analice las consultas lentas y busque técnicas de optimización.
  - En las cargas de trabajo de ElastiCache para Valkey y Redis OSS, obtendrá más información sobre sus consultas mediante el análisis del registro lento. Por ejemplo, puede utilizar el siguiente comando, `valkey-cli slowlog get 10`, para mostrar los últimos 10 comandos que han superado los umbrales de latencia (10 milisegundos de forma predeterminada).
  - Algunas consultas se pueden realizar de forma más eficiente con estructuras de datos complejas de ElastiCache para Valkey y Redis OSS. Por ejemplo, en el caso de las búsquedas de rangos de estilos numéricos, una aplicación puede implementar índices numéricos simples con conjuntos ordenados. La administración de estos índices puede reducir los escaneos realizados en el conjunto de datos y devolver los datos con una mayor eficiencia de rendimiento.
  - Para las cargas de trabajo de ElastiCache para Valkey y Redis OSS, `redis-benchmark` proporciona una interfaz sencilla para probar el rendimiento de diferentes comandos mediante entradas definidas por el usuario, como la cantidad de clientes y el tamaño de los datos.
  - Dado que Memcached solo admite comandos simples a nivel de clave, considere la posibilidad de crear claves adicionales como índices para evitar iterar a través del espacio de claves para atender las consultas de los clientes.
- [Recursos]:
  - Monitoreo del uso con métricas de CloudWatch
  - Uso de las alarmas de Amazon CloudWatch
  - Parámetros específicos de Valkey y Redis OSS
  - SLOWLOG
  - benchmark

PE 2: ¿Cómo se distribuye el trabajo entre los nodos del clúster de ElastiCache?

Introducción a nivel de pregunta: La forma en que la aplicación se conecta a los nodos de Amazon ElastiCache puede influir en el rendimiento y la escalabilidad del clúster.

Ventaja a nivel de pregunta: Al utilizar de forma adecuado los nodos disponibles en el clúster garantizará que el trabajo se distribuya entre los recursos disponibles. Las siguientes técnicas también ayudan a evitar que haya recursos inactivos.

- [Obligatorio] Haga que los clientes se conecten al punto de conexión de ElastiCache adecuado.
  - ElastiCache para Valkey y Redis OSS implementa diferentes puntos de conexión en función del modo de clúster en uso. Si el modo de clúster está habilitado, ElastiCache proporcionará un punto de conexión de configuración. Para el modo de clúster deshabilitado, ElastiCache proporciona un punto de conexión principal, que normalmente se usa para escritura, y un punto de conexión de lectura para equilibrar las lecturas entre las réplicas. La implementación correcta de estos puntos de conexión resultará en un mejor rendimiento y en operaciones de escalado más sencillas. Evite conectarse a puntos de conexión de nodos individuales a menos que haya un requisito específico para hacerlo.
  - Para los clústeres de Memcached de varios nodos, ElastiCache proporciona un punto de conexión de configuración que permite la detección automática. Se recomienda utilizar un algoritmo de hash para distribuir el trabajo de manera uniforme entre los nodos de caché. Muchas bibliotecas cliente de Memcached implementan un hash coherente. Consulte la documentación de la biblioteca que va a utilizar para ver si admite el uso consistente de hash y saber cómo implementarlo. Encontrará más información sobre la implementación de estas funciones aquí.
- [Mejor] Aproveche el modo de clúster habilitado de ElastiCache para Valkey y Redis OSS para mejorar la escalabilidad.
  - Los clústeres de ElastiCache para Valkey y Redis OSS (modo de clúster habilitado) admiten operaciones de escalado en línea (escalar/reducir horizontalmente y verticalmente) para ayudar a distribuir los datos de forma dinámica entre las particiones. El uso del punto de conexión de configuración garantizará que los clientes compatibles con clústeres puedan adaptarse a los cambios en la topología del clúster.
  - También puede reequilibrar el clúster al mover las ranuras hash entre las particiones disponibles en su clúster de ElastiCache para Valkey y Redis OSS (modo de clúster habilitado). Esto ayuda a distribuir el trabajo de manera más eficiente entre las particiones disponibles.

- [Mejor] Implemente una estrategia para identificar y corregir las teclas de acceso rápido de su carga de trabajo.
  - Tenga en cuenta el impacto de las estructuras de datos multidimensionales de Valkey o Redis OSS, como listas, flujos, conjuntos, etc. Estas estructuras de datos se almacenan en claves únicas, que residen en un solo nodo. Una clave multidimensional muy grande tiene el potencial de utilizar más capacidad de red y memoria que otros tipos de datos y puede provocar un uso desproporcionado de ese nodo. Si es posible, diseñe la carga de trabajo de modo que distribuya el acceso a los datos entre varias claves discretas.
  - Las teclas de acceso rápido de la carga de trabajo pueden influir en el rendimiento del nodo en uso. Para las cargas de trabajo de ElastiCache para Valkey y Redis OSS, puede detectar las teclas de acceso rápido mediante `valkey-cli --hotkeys` si existe una política de memoria máxima de LFU.
  - Considere la posibilidad de replicar las teclas de acceso rápido en varios nodos para distribuir el acceso a ellas de manera más uniforme. Este enfoque requiere que el cliente escriba en varios nodos principales (el nodo de Valkey o Redis OSS en sí no proporciona esta funcionalidad) y mantenga una lista de nombres de claves de la que leer, además del nombre de la clave original.
  - El motor de ElastiCache 7.2 para Valkey 7.2 y ElastiCache versión 6 para Redis OSS y versiones posteriores admiten el almacenamiento en caché en el cliente asistido por el servidor. Esto permite que las aplicaciones esperen a que se modifique una clave antes de volver a realizar llamadas de red a ElastiCache.
- [Recursos]:
  - Configure ElastiCache for Valkey and Redis OSS for higher availability
  - Búsqueda de puntos de conexión en ElastiCache
  - Procedimientos recomendados para el equilibrio de cargas
  - Cambio de particiones en línea para Valkey o Redis OSS (modo de clúster habilitado)
  - Client-side caching in Valkey and Redis OSS

PE 3: En el caso del almacenamiento en caché de las cargas de trabajo, ¿cómo se realiza el seguimiento de la eficacia y el rendimiento de la memoria caché y se informa al respecto?

Introducción a nivel de pregunta: El almacenamiento en caché es una carga de trabajo habitual en ElastiCache y es importante que comprenda cómo administrar la eficacia y el rendimiento de la memoria caché.

Ventaja a nivel de pregunta: Es posible que su aplicación muestre signos de un rendimiento lento. La capacidad de utilizar métricas específicas de la memoria caché para tomar una decisión sobre cómo aumentar el rendimiento de la aplicación es fundamental para la carga de trabajo de la memoria caché.

- [Obligatorio] Mida y realice un seguimiento a lo largo del tiempo de la proporción de aciertos de la caché. La eficiencia de la memoria caché está determinada por su proporción de aciertos de la caché. La proporción de aciertos de la caché es el total de aciertos de caché dividido por el total de aciertos y errores. Cuanto más se acerque a 1 esté la proporción, más eficaz será la memoria caché. Una baja proporción de aciertos de caché se debe al volumen de errores de caché. Los errores de caché se producen cuando la clave solicitada no se encuentra en la memoria caché. Una clave no está en la memoria caché porque ha sido expulsada o eliminada, ha caducado o no ha existido nunca. Comprenda por qué las claves no están en la memoria caché y desarrolle las estrategias adecuadas para tenerlas en la memoria caché.

  [Recursos]:

  - Métricas de Valkey y Redis OSS
- [Obligatorio] Mida y recopile el rendimiento de la caché de su aplicación junto con los valores de latencia y uso de la CPU para saber si necesita realizar ajustes en el tiempo de vida o en otros componentes de la aplicación. ElastiCache proporciona un conjunto de métricas de CloudWatch de las latencias agregadas para cada estructura de datos. Estas métricas de latencia se calculan mediante la estadística commandstats del comando INFO y no incluyen el tiempo de red ni el tiempo de E/S. Se trata solo del tiempo que consume ElastiCache para procesar las operaciones.

  [Recursos]:

  - Métricas de Valkey y Redis OSS
  - Monitoring best practices with ElastiCache using Amazon CloudWatch
- [Lo mejor] Elija la estrategia de almacenamiento en caché adecuada para sus necesidades. Una baja proporción de aciertos de caché se debe al volumen de errores de caché. Si su carga

de trabajo está diseñada para tener un bajo volumen de errores de caché (como comunicación en tiempo real), es mejor revisar las estrategias de almacenamiento en caché y aplicar las resoluciones más adecuadas para la carga de trabajo, como la instrumentación de consultas para medir la memoria y el rendimiento. Las estrategias reales que implemente para completar y mantener su caché dependen del tipo de datos que sus clientes necesiten almacenar en la caché, así como de los patrones de acceso a dichos datos. Por ejemplo, es poco probable que utilice la misma estrategia tanto para las recomendaciones personalizadas en una aplicación de streaming como para las noticias más populares.

[Recursos]:

- Estrategias de almacenamiento en caché para Memcached
- Caching Best Practices
- Documento técnico Performance at Scale with Amazon ElastiCache

PE 4: ¿Cómo optimiza la carga de trabajo el uso de los recursos y las conexiones de red?

Introducción en la pregunta: Muchos clientes de aplicaciones admiten ElastiCache para Valkey, Memcached y Redis OSS. Además, las implementaciones pueden variar. Debe comprender la administración de redes y conexiones implementada para analizar el posible impacto en el rendimiento.

Ventaja a nivel de pregunta: El uso eficiente de los recursos de red puede mejorar la eficiencia del rendimiento de su clúster. Las siguientes recomendaciones pueden reducir las demandas de red y mejorar la latencia y el rendimiento del clúster.

- [Obligatorio] Gestione de forma proactiva las conexiones a su clúster de ElastiCache.
  - La agrupación de conexiones en la aplicación reduce la sobrecarga del clúster que se crea al abrir y cerrar conexiones. Supervise el comportamiento de la conexión en Amazon CloudWatch mediante `CurrConnections` y `NewConnections`.
  - Cierre correctamente las conexiones del cliente cuando corresponda para evitar fugas de conexiones. Las estrategias de administración de conexiones incluyen cerrar correctamente las conexiones que no están en uso y establecer tiempos de espera de conexión.
  - Para las cargas de trabajo de Memcached, hay una cantidad configurable de memoria reservada para gestionar las conexiones denominada `memcached_connections_overhead`.
- [Mejor] Comprima objetos grandes para reducir la memoria y mejorar el rendimiento de la red.

- La compresión de datos puede reducir la cantidad de rendimiento de red requerida (Gbps), pero aumenta la cantidad de trabajo de la aplicación para comprimir y descomprimir datos.
- La compresión también reduce la cantidad de memoria que consumen las claves
- En función de las necesidades de su aplicación, considere las ventajas y desventajas entre la relación y la velocidad de compresión.

- [Recursos]:
  - ElastiCache - Global Datastore
  - Parámetros específicos de Memcached
  - ElastiCache version 5.0.3 for Redis OSS enhances I/O handling to boost performance
  - Métricas de Valkey y Redis OSS
  - Configure ElastiCache for higher availability

PE 5: ¿Cómo se gestiona la eliminación o la expulsión de claves?

Introducción a nivel de pregunta: Las cargas de trabajo tienen diferentes requisitos, y un comportamiento esperado cuando un nodo del clúster se acerca a los límites de consumo de memoria. ElastiCache cuenta con diferentes políticas para gestionar estas situaciones.

Ventaja a nivel de pregunta: La gestión adecuada de la memoria disponible y la comprensión de las políticas de expulsión ayudarán a garantizar el conocimiento del comportamiento del clúster cuando se superen los límites de memoria de las instancias.

- [Obligatorio] Analice el acceso a los datos para evaluar qué política aplicar. Identifique una política de memoria máxima adecuada para controlar si se realizan expulsiones en el clúster y cómo se hacen.
  - La expulsión tiene lugar cuando se consume la cantidad máxima de memoria del clúster y existe una política que permite la expulsión. El comportamiento del clúster en esta situación depende de la política de expulsión especificada. Esta política se puede administrar con la `maxmemory-policy` en el grupo de parámetros del clúster.
  - La política predeterminada `volatile-lru` libera memoria al expulsar las claves con una fecha de caducidad establecida (valor de TTL). Las políticas menos usadas frecuentemente (LFU) y menos usadas recientemente (LRU) eliminan las claves en función del uso.
  - Para las cargas de trabajo de Memcached, existe una política LRU predeterminada que controla las expulsiones en cada nodo. Es posible supervisar el número de expulsiones de su clúster de Amazon ElastiCache mediante la métrica de expulsiones de Amazon CloudWatch.

- [Mejor] Estandarice el comportamiento de eliminación para controlar el impacto en el rendimiento de su clúster y evitar atascos inesperados en el rendimiento.
  - En el caso de las cargas de trabajo de ElastiCache para Valkey y Redis OSS, al eliminar claves del clúster de forma explícita, `UNLINK` es como `DEL`, pues elimina claves especificadas. Sin embargo, el comando realiza la recuperación de memoria real en un subproceso diferente, por lo que no es de bloqueo, mientras que `DEL` sí. La eliminación real se realizará más adelante de forma asincrónica.
  - Para las cargas de trabajo de ElastiCache versión 6.x para Redis OSS, el comportamiento del comando `DEL` se puede modificar en el grupo de parámetros mediante el parámetro `lazyfree-lazy-user-del`.
- [Recursos]:
  - Configuración de los parámetros de motor mediante grupos de parámetros de ElastiCache
  - UNLINK
  - Administración financiera en la nube con AWS

PE 6: ¿Cómo se modelan los datos y se interactúa con ellos en ElastiCache?

Introducción a nivel de pregunta: ElastiCache depende en gran medida de las aplicaciones en las estructuras de datos y modelo de datos utilizados, pero también debe tener en cuenta el almacén de datos subyacente (si está presente). Conozca las estructuras de datos disponibles y asegúrese de utilizar las estructuras de datos más adecuadas para sus necesidades.

Ventaja a nivel de pregunta: El modelado de datos en ElastiCache tiene varias capas, que incluyen los casos de uso de la aplicación, los tipos de datos y las relaciones entre los elementos de datos. Además, cada comando y tipo de datos tiene sus propias firmas de rendimiento bien documentadas.

- [Lo mejor] Una práctica recomendada es reducir la sobrescritura involuntaria de datos. Utilice una convención de nomenclatura que minimice la superposición de nombres de clave. La nomenclatura convencional de las estructuras de datos utiliza un método jerárquico como: `APPNAME:CONTEXT:ID`, por ejemplo `ORDER-APP:CUSTOMER:123`.

  [Recursos]:
  - Key naming
- [Lo mejor] Los comandos de ElastiCache para Valkey y Redis OSS tienen una complejidad temporal definida por la notación Big O. Esta complejidad temporal de un comando es una representación algorítmica/matemática de su impacto. Al introducir un nuevo tipo de datos en su

aplicación, debe revisar detenidamente la complejidad temporal de los comandos relacionados. Los comandos con una complejidad temporal de O(1) son constantes y no dependen de la cantidad de datos de entrada.; sin embargo, los comandos con una complejidad temporal de O(N) son lineales y están sujetos a la cantidad de datos de entrada. Debido a que ElastiCache para Valkey y Redis OSS se ha diseñado con un único subproceso, un gran volumen de operaciones de alta complejidad temporal se traducirá en un menor rendimiento y en posibles tiempos de espera de las operaciones.

[Recursos]:

- Comandos

- [Lo mejor] Utilice las API para obtener visibilidad de la GUI en el modelo de datos de su clúster.

  [Recursos]:

  - Redis OSS Commander
  - Redis OSS Browser
  - Redsmin

PE 7: ¿Cómo se registran los comandos de ejecución lenta en el clúster de Amazon ElastiCache?

Introducción a nivel de pregunta: Ventajas del ajuste del rendimiento mediante la captura, la agregación y la notificación de comandos de ejecución prolongada. Si comprende cuánto tiempo tardan en ejecutarse los comandos, puede determinar qué comandos tienen un rendimiento deficiente y qué comandos impiden que el motor funcione de manera óptima. ElastiCache también tiene la capacidad de enviar esta información a Amazon CloudWatch o Amazon Kinesis Data Firehose.

Ventaja a nivel de pregunta: El registro en una ubicación permanente dedicada y la provisión de eventos de notificación para los comandos lentos puede ayudar a realizar un análisis detallado del rendimiento y se puede utilizar para activar eventos automatizados.

- [Obligatorio] ElastiCache con una versión del motor Valkey 7.2 o posterior, o un motor de Redis OSS versión 6.0 o posterior, un grupo de parámetros configurado correctamente y el registro SLOWLOG habilitado en el clúster.
  - Los parámetros requeridos solo están disponibles cuando la compatibilidad de la versión del motor está configurada para Valkey 7.2 y versiones posteriores o Redis OSS versión 6.0 o posteriores.

  - El registro SLOWLOG se produce cuando el tiempo de ejecución de un comando en el servidor supera un valor especificado. El comportamiento del clúster depende de los parámetros del grupo de parámetros asociado, que son `slowlog-log-slower-than` y `slowlog-max-len`.
  - Los cambios surten efecto inmediatamente.
- [Lo mejor] Aproveche las capacidades de CloudWatch o Kinesis Data Firehose.
  - Utilice las capacidades de filtrado y alarma de CloudWatch, CloudWatch Logs Insights y Amazon Simple Notification Services para supervisar el rendimiento y notificar eventos.
  - Utilice las capacidades de transmisión de Kinesis Data Firehose para archivar los registros SLOWLOG en un almacenamiento permanente o para activar el ajuste automático de los parámetros del clúster.
  - Determine si el formato JSON o TEXTO sin formato se adapta mejor a sus necesidades.
  - Proporcione permisos de IAM para publicar en CloudWatch o Kinesis Data Firehose.
- [Mejor] Configure `slowlog-log-slower-than` con un valor distinto al predeterminado.
  - Este parámetro determina cuánto tiempo puede ejecutarse un comando en el motor de Valkey o Redis OSS antes de que se registre como un comando de ejecución lenta. El valor predeterminado es 10 000 microsegundos (10 milisegundos). El valor predeterminado puede ser demasiado alto para algunas cargas de trabajo.
  - Determine un valor que sea más adecuado para su carga de trabajo en función de las necesidades de la aplicación y los resultados de las pruebas; sin embargo, un valor demasiado bajo puede generar un exceso de datos.
- [Mejor] Deje `slowlog-max-len` como valor predeterminado.
  - Este parámetro determina el límite superior del número de comandos de ejecución lenta que se capturan en la memoria de Valkey o Redis OSS en un momento dado. Un valor de 0 desactiva de manera efectiva la captura. Cuanto más alto sea el valor, más entradas se almacenarán en la memoria, lo que reducirá la posibilidad de que se expulse información importante antes de poder revisarla. El valor predeterminado es 128.
  - El valor predeterminado es adecuado para la mayoría de las cargas de trabajo. Si es necesario analizar los datos en un período de tiempo ampliado desde valkey-cli mediante el comando SLOWLOG, considere aumentar este valor. Esto permite que queden más comandos en la memoria de Valkey o Redis OSS.

    Si emite datos de SLOWLOG a Registros de CloudWatch o Kinesis Data Firehose, los datos se conservarán y se podrán analizar fuera del sistema de ElastiCache, lo que reducirá la necesidad

de almacenar grandes cantidades de comandos de ejecución lenta en la memoria de Valkey o Redis OSS.

- [Recursos]:
  - How do I turn on Slow log in a cluster?
  - Entrega de registro
  - Parámetros específicos de Redis OSS
  - https://aws.amazon.com/cloudwatch/Amazon CloudWatch
  - Amazon Kinesis Data Firehose

PE8: ¿Cómo ayuda el escalado automático a aumentar el rendimiento del clúster de ElastiCache?

Introducción de pregunta: al implementar la característica de escalado automático de Valkey o Redis OSS, los componentes de ElastiCache pueden ajustarse con el tiempo para que aumenten o disminuyan automáticamente las particiones o réplicas deseadas. Esto se puede hacer mediante la implementación de políticas de escalado programado o de seguimiento de objetivos.

Ventaja a nivel de pregunta: Comprender y planificar los picos de carga de trabajo puede garantizar un mejor rendimiento del almacenamiento en caché y la continuidad empresarial. El escalado automático de ElastiCache supervisa continuamente el uso de la CPU/memoria a fin de garantizar que el clúster funcione a los niveles de rendimiento deseados.

- [Obligatorio] Al lanzar un clúster para ElastiCache para Valkey o Redis OSS:
  1. Asegúrese de que el modo Clúster esté habilitado.
  2. Asegúrese de que la instancia pertenezca a una familia de un cierto tipo y tamaño que admita el escalado automático.
  3. Asegúrese de que el clúster no se ejecute en almacenes de datos globales, Outposts o zonas locales.

  [Recursos]:
  - Escalado de clústeres en Valkey y Redis OSS (modo de clúster habilitado)
  - Uso de Auto Scaling con particiones
  - Uso de Auto Scaling con réplicas
- [Lo mejor] Identifique si su carga de trabajo requiere mucha lectura o escritura para definir la política de escalado. Para obtener el mejor rendimiento, utilice solo una métrica de seguimiento.

Se recomienda evitar tener varias políticas para cada dimensión, ya que las políticas de escalado automático se escalan horizontalmente de forma gradual cuando se alcanza el objetivo, pero solo se reducen horizontalmente cuando todas las políticas de seguimiento de objetivos están listas para reducirse horizontalmente.

[Recursos]:

- Políticas de Auto Scaling
- Defining a scaling policy

- [Lo mejor] Supervisar el rendimiento a lo largo del tiempo puede ayudar a detectar cambios en la carga de trabajo que pasarían desapercibidos si se hiciera en un momento determinado. Puede analizar las métricas correspondientes de CloudWatch para la utilización del clúster durante un período de cuatro semanas a fin de determinar el umbral del valor objetivo. Si aún no está seguro de qué valor elegir, se recomienda comenzar con el valor mínimo admitido de métrica predefinida.

  [Recursos]:

  - Monitoreo del uso con métricas de CloudWatch
- [Mejor] Se recomienda probar la aplicación con las cargas de trabajo mínimas y máximas esperadas para identificar la cantidad exacta de particiones o réplicas que necesita el clúster a fin de desarrollar políticas de escalado y mitigar los problemas de disponibilidad.

  [Recursos]:

  - Registro de un destino escalable
  - Registering a Scalable Target using the AWS CLI

Pilar de optimización de costos del enfoque Well-Architected de Amazon ElastiCache

El pilar de optimización de costes se centra en evitar costos innecesarios. Los temas clave incluyen comprender y controlar dónde se gasta el dinero, seleccionar el tipo de nodo más adecuado (utilice instancias que admitan la organización de datos en niveles en función de las necesidades de la carga de trabajo), el número correcto de tipos de recursos (el número de réplicas leídas), analizar el gasto a lo largo del tiempo y escalar para satisfacer las necesidades empresariales sin gastar de más.

Temas

- COST 1: ¿Cómo se identifica y se realiza el seguimiento de los costos asociados a los recursos de ElastiCache? ¿Cómo se desarrollan los mecanismos que permiten a los usuarios crear, gestionar y eliminar los recursos creados?
- COST 2: ¿Cómo se utilizan las herramientas de supervisión continua para ayudar a optimizar los costos asociados a los recursos de ElastiCache?
- COST 3: ¿Debe usar un tipo de instancia que admita la organización de datos en niveles? ¿Cuáles son las ventajas de la organización de datos en niveles? ¿Cuándo no se deben usar instancias de almacenamiento de datos en niveles?

COST 1: ¿Cómo se identifica y se realiza el seguimiento de los costos asociados a los recursos de ElastiCache? ¿Cómo se desarrollan los mecanismos que permiten a los usuarios crear, gestionar y eliminar los recursos creados?

Introducción a nivel de pregunta: Para comprender las métricas de costos es preciso la participación y la colaboración de varios equipos: ingeniería de software, gestión de datos, propietarios de productos, finanzas y liderazgo. Identificar los principales factores de los costos requiere que todas las partes involucradas comprendan las herramientas de control del uso de los servicios y las soluciones de compromiso que conlleva la administración de costos. Con frecuencia es la diferencia clave entre los esfuerzos de optimización de costos fructíferos e infructuosos. Garantizar que cuente con procesos y herramientas para realizar un seguimiento de los recursos creados desde el desarrollo hasta la producción y la retirada le ayudara a gestionar los costos asociados a ElastiCache.

Ventaja a nivel de pregunta: El seguimiento continuo de todos los costos asociados a la carga de trabajo requiere un conocimiento profundo de la arquitectura que incluye ElastiCache como uno de sus componentes. Además, debe contar con un plan de administración de costos para recopilar y comparar el uso con su presupuesto.

- [Obligatorio] Establezca un centro de excelencia en la nube (CCoE) que tenga un estatuto constitutivo que se encargue de definir, seguir y tomar medidas relacionado con las métricas en torno al uso de ElastiCache en su organización. Si hay un CCoE y funciona, asegúrese de que sepa cómo interpretar y seguir los costos asociados con ElastiCache. Cuando se creen recursos, utilice políticas y roles de IAM para validar que solo equipos y grupos específicos puedan crear instancias de los recursos. Esto garantiza que los costos estén asociados a los resultados comerciales y que se establezca una línea clara de responsabilidad, desde la perspectiva de los costos.

1. El CCoE debe identificar, definir y publicar métricas de costos que se actualicen de forma periódica (mensual) en torno al uso clave de ElastiCache en datos categóricos, como:
   a. Tipos de nodos utilizados y sus atributos: instancias estándar frente a instancias optimizadas para memoria, instancias bajo demanda o reservadas, regiones y zonas de disponibilidad
   b. Tipos de entornos: gratuitos, de desarrollo, de pruebas y de producción
   c. Estrategias de almacenamiento y retención de copias de seguridad
   d. Transferencia de datos dentro y entre regiones
   e. Instancias que se ejecutan en Amazon Outposts
2. El CCoE está formado por un equipo multifuncional con representación no exclusiva de los equipos de ingeniería de software, gestión de datos, equipo de productos, finanzas y liderazgo de su organización.

[Recursos]:

- Create a Cloud Center of Excellence
- Precios de Amazon ElastiCache

- [Obligatorio] Utilice etiquetas de asignación de costos para realizar un seguimiento de los costos con un nivel de detalle bajo. Utilice la administración de costos de AWS para visualizar, comprender y administrar los costos de AWS y el uso a lo largo del tiempo.
  1. Puede utilizar etiquetas para organizar los recursos y etiquetas de asignación de costos para realizar un seguimiento de los costos de AWS en un nivel detallado. Después de activar las etiquetas de asignación de costos, AWS las utiliza para organizar los costos de los recursos en el informe de asignación de costos, de forma que le resulte más fácil clasificar en categorías los costos de AWS y hacer un seguimiento de ellos. AWS proporciona dos tipos de etiquetas de asignación de costos, las etiquetas de generadas por AWS y las etiquetas definidas por el usuario. AWS define, crea y aplica las etiquetas generadas por AWS por usted, mientras usted define, crea y aplica las etiquetas definidas por el usuario. Debe activar ambos tipos de etiquetas por separado para que puedan aparecer en Cost Management o en un informe de asignación de costos.
  2. Utilice las etiquetas de asignación de costos para organizar la factura de AWS de modo que refleje su propia estructura de costos. Al agregar etiquetas de asignación de costos a sus recursos en Amazon ElastiCache, podrá realizar un seguimiento de los costos mediante el agrupamiento de los gastos en sus facturas por valores de etiqueta de recursos. Plantéese combinar etiquetas para realizar un seguimiento de los costos con un mayor nivel de detalle.

[Recursos]:

  - Uso de etiquetas de asignación de costos de AWS
  - Monitoreo de costos con etiquetas de asignación de costos
  - AWS Explorador de costos de
- [Lo mejor] Conecte el costo de ElastiCache con las métricas que llegan a toda la organización.
  1. Tenga en cuenta las métricas empresariales y las métricas operativas, como la latencia: ¿qué conceptos de su modelo de negocio son comprensibles en todos los roles? Las métricas deben ser comprensibles para el mayor número posible de roles de la organización.
  2. Ejemplos: usuarios atendidos simultáneamente, latencia máxima y promedio por operación y usuario, puntuaciones de participación de los usuarios, tasas de retorno de usuarios por semana, duración de la sesión por usuario, tasa de abandono, tasa de aciertos de caché y seguimiento de las claves.

  [Recursos]:

  - Monitoreo del uso con métricas de CloudWatch
- [Bueno] Mantenga una visibilidad operativa y arquitectónica actualizada de las métricas y los costos de toda la carga de trabajo que usa ElastiCache.
  1. Entienda todo su ecosistema de soluciones: ElastiCache suele formar parte de un ecosistema completo de servicios de AWS en su conjunto tecnológico, desde clientes hasta API Gateway, Redshift y QuickSight para las herramientas de generación de informes (por ejemplo).
  2. Mapee los componentes de su solución, desde los clientes, las conexiones, la seguridad, las operaciones en memoria, el almacenamiento, la automatización de recursos, el acceso y la administración de datos, en su diagrama de arquitectura. Cada capa se conecta a toda la solución y tiene sus propias necesidades y capacidades que aumentan el costo total o ayudan a gestionarlo.
  3. El diagrama debe incluir el uso de políticas de computación, redes, almacenamiento, ciclo de vida y recopilación de métricas, así como los elementos operativos y funcionales de ElastiCache de su aplicación.
  4. Es probable que los requisitos de la carga de trabajo evolucionen con el tiempo por lo que es esencial que siga manteniendo y documentando su comprensión de los componentes subyacentes, así como de sus objetivos funcionales principales, a fin de mantener una actitud proactiva en la gestión de los costes de la carga de trabajo.
  5. El apoyo ejecutivo respecto a la visibilidad, la responsabilidad, la priorización y los recursos es crucial para tener una estrategia de administración de costos eficaz para su ElastiCache.

COST 2: ¿Cómo se utilizan las herramientas de supervisión continua para ayudar a optimizar los costos asociados a los recursos de ElastiCache?

Introducción a nivel de pregunta: Debe buscar un equilibrio adecuado entre las métricas de costos de ElastiCache y de rendimiento de la aplicación. Amazon CloudWatch proporciona visibilidad de las métricas operativas clave que pueden ayudarlo a evaluar si sus recursos de ElastiCache están sobreutilizados o infrautilizados, en relación con sus necesidades. Desde el punto de vista de la optimización de costos, debe comprender cuándo está sobreaprovisionado y poder desarrollar los mecanismos adecuados para cambiar el tamaño de sus recursos de ElastiCache y, al mismo tiempo, mantener sus necesidades operativas, de disponibilidad, de resiliencia y de rendimiento.

Ventaja a nivel de pregunta: En un estado ideal, habrá aprovisionado suficientes recursos para satisfacer las necesidades operativas de su carga de trabajo y no tendrá recursos infrautilizados que puedan llevar a un estado de costos deficiente. Debe poder identificar y evitar utilizar recursos de ElastiCache sobredimensionados durante períodos prolongados.

- [Obligatorio] Utilice CloudWatch para supervisar sus clústeres de ElastiCache y analizar cómo se relacionan estas métricas con sus paneles de AWS Cost Explorer.
  1. ElastiCache proporciona métricas de nivel de anfitrión (por ejemplo, sobre el uso de la CPU) y métricas específicas del software del motor de caché (por ejemplo, las tasas de caché y los errores de caché). Estas métricas se miden y publican para cada nodo de caché en intervalos de 60 segundos.
  2. Las métricas de rendimiento de ElastiCache (CPUUtilization, EngineUtilization, SwapUsage, CurrConnections y Evictions) pueden indicar que necesita escalar o reducir verticalmente o (utilizar tipos de nodos de caché más grandes o más pequeños) o escalar o reducir horizontalmente (agregar más/menos particiones). Comprenda las implicaciones financieras de las decisiones de escalado mediante la creación de una matriz de estrategias que estime el costo adicional y los períodos de tiempo mínimos y máximos necesarios para cumplir con los umbrales de rendimiento de sus aplicaciones.

  [Recursos]:

  - Monitoreo del uso con métricas de CloudWatch
  - Qué métricas debo monitorear?
  - Precios de Amazon ElastiCache
- [Obligatorio] Comprenda y documente su estrategia de copias de seguridad y las implicaciones de costos.

1. Con ElastiCache, las copias de seguridad se almacenan en Amazon S3, que proporciona un almacenamiento duradero. Debe comprender las implicaciones de costos en relación con su capacidad para recuperarse en caso de errores.
2. Habilite copias de seguridad automáticas que eliminarán los archivos de copia de seguridad que hayan superado el límite de retención.

[Recursos]:

- Programación de copias de seguridad automáticas
- Precios de Amazon Simple Storage Service

- [Lo mejor] Utilice nodos reservados para sus instancias como estrategia deliberada para administrar los costos de las cargas de trabajo que se conocen bien y están documentadas. Los nodos reservados se cobran por adelantado en función de una tarifa que depende del tipo de nodo y la duración de la reserva: uno o tres años. Dicho cargo es mucho menor que el cargo por uso por hora que se aplica con los nodos bajo demanda.
  1. Es posible que tenga que operar sus clústeres de ElastiCache mediante nodos bajo demanda hasta que haya recopilado suficientes datos para estimar los requisitos de instancias reservadas. Planifique y documente los recursos necesarios para satisfacer sus necesidades y compare los costos esperados entre los tipos de instancias (bajo demanda o reservadas).
  2. Evalúe periódicamente los nuevos tipos de nodos de caché disponibles y determine si tiene sentido, desde una perspectiva de métricas operativas y de costos, migrar su flota de instancias a nuevos tipos de nodos de caché.

COST 3: ¿Debe usar un tipo de instancia que admita la organización de datos en niveles? ¿Cuáles son las ventajas de la organización de datos en niveles? ¿Cuándo no se deben usar instancias de almacenamiento de datos en niveles?

Introducción a nivel de pregunta: Seleccionar el tipo de instancia adecuado no solo puede tener un impacto en el rendimiento y el nivel de servicio, sino también en los aspectos financieros. Los tipos de instancias tienen diferentes costos asociados. Una decisión natural es seleccionar uno o varios tipos de instancias grandes que puedan adaptarse a todas las necesidades de almacenamiento de memoria. Sin embargo, esto podría tener un impacto significativo en los costos a medida que el proyecto madure. Para asegurarse de seleccionar el tipo de instancia correcto, es necesario examinar periódicamente el tiempo de inactividad del objeto ElastiCache.

Ventaja a nivel de pregunta: Debe tener una idea clara de cómo los distintos tipos de instancias afectan a sus costes en el presente y en el futuro. Los cambios marginales o periódicos en la carga

de trabajo no deberían provocar cambios desproporcionados en los costos. Si la carga de trabajo lo permite, los tipos de instancias que admiten la organización de datos en niveles ofrecen un mejor precio por almacenamiento disponible. Gracias al almacenamiento SSD disponible por instancia, las instancias de organización de datos en niveles admiten una capacidad total de datos por instancia mucho mayor.

- [Obligatorio] Comprenda las limitaciones de las instancias de almacenamiento de datos en niveles.
  1. Solo están disponibles para clústeres de ElastiCache para Valkey o Redis OSS.
  2. Solo algunos tipos de instancias admiten la organización de datos en niveles.
  3. Solo se admite la versión 6.2 y superior de ElastiCache para Redis OSS.
  4. Los elementos grandes no se intercambian a SSD. Los objetos de más de 128 MiB se guardan en la memoria.

  [Recursos]:

  - Organización de datos en niveles
  - Precios de Amazon ElastiCache
- [Obligatorio] Comprenda a qué porcentaje de su base de datos accede regularmente su carga de trabajo.
  1. Las instancias de almacenamiento de datos en niveles son ideales para cargas de trabajo que, a menudo, acceden a una pequeña parte del conjunto de datos general, pero que aun así requieren un acceso rápido al resto de los datos. En otras palabras, la relación entre datos calientes y templados es de aproximadamente 20:80.
  2. Desarrolle un seguimiento a nivel de clúster del tiempo de inactividad de los objetos.
  3. Las implementaciones grandes de más de 500 Gb de datos son buenas candidatas.
- [Obligatorio] Comprenda que las instancias de organización de datos en niveles no son opcionales para determinadas cargas de trabajo.
  1. El acceso a los objetos que se utilizan con menos frecuencia conlleva un pequeño costo de rendimiento, ya que se intercambian a SSD locales. Si su aplicación depende del tiempo de respuesta, pruebe el impacto en su carga de trabajo.
  2. No resulta adecuado para cachés que almacenan principalmente objetos grandes de más de 128 MiB de tamaño.

  [Recursos]:

  - Limitaciones

- [Lo mejor] Los tipos de instancias reservadas admiten la organización de datos en niveles. Esto garantiza un menor costo en términos de cantidad de almacenamiento de datos por instancia.
  1. Es posible que tenga que operar sus clústeres de ElastiCache mediante instancias sin organización de datos en niveles hasta que comprenda mejor sus requisitos.
  2. Analice el patrón de uso de datos de sus clústeres de ElastiCache.
  3. Cree un trabajo automatizado que recopile periódicamente el tiempo de inactividad de los objetos.
  4. Si observa que un gran porcentaje (alrededor del 80 %) de los objetos permanecen inactivos durante un período de tiempo que considere apropiado para su carga de trabajo, documente los hallazgos y sugiera migrar el clúster a instancias que admitan la organización de datos en niveles.
  5. Evalúe periódicamente los nuevos tipos de nodos de caché disponibles y determine si tiene sentido, desde una perspectiva de métricas operativas y de costos, migrar su flota de instancias a nuevos tipos de nodos de caché.

  [Recursos]:

  - OBJECT IDLETIME
  - Precios de Amazon ElastiCache

Prácticas recomendadas y pasos comunes de solución de problemas con ElastiCache

Los siguientes temas le proporcionan consejos para solucionar errores y problemas que pueda encontrar al utilizar ElastiCache. Si se encuentra con un problema que no se muestra aquí, puede utilizar el botón de feedback de esta página para notificarlo.

Para obtener más consejos sobre la resolución de problemas y respuestas a preguntas comunes de compatibilidad, visite el Centro de conocimiento de AWS.

Temas

- Problemas de conectividad
- Errores del cliente de Valkey o Redis OSS
- Solución de problemas de alta latencia en ElastiCache sin servidor
- Solución de problemas de limitación en ElastiCache sin servidor

- Problemas de las conexiones persistentes
- Temas relacionados

Problemas de conectividad

Si no puede conectarse a la caché de ElastiCache, considere una de las siguientes opciones:

1. Uso de TLS: si se produce un bloqueo de la conexión al punto de conexión de ElastiCache, es posible que no esté utilizando TLS en su cliente. Si utiliza ElastiCache sin servidor, el cifrado en tránsito siempre está habilitado. Asegúrese de que su cliente utilice TLS para conectarse a la caché. Obtenga más información sobre cómo conectarte a una caché con TLS habilitado.
2. VPC: solo se puede acceder a las cachés de ElastiCache desde una VPC. Asegúrese de que la instancia de EC2 desde la que accede a la caché y la caché de ElastiCache se creen en la misma VPC. Como alternativa, debe habilitar el emparejamiento de VPC entre la VPC en la que reside la instancia de EC2 y la VPC en la que va a crear la caché.
3. Grupos de seguridad: ElastiCache utiliza grupos de seguridad para controlar el acceso a la caché. Considere lo siguiente:
   a. Asegúrese de que el grupo de seguridad utilizado por la caché de ElastiCache permita el acceso entrante desde la instancia de EC2. Consulte aquí cómo configurar correctamente las reglas de entrada en su grupo de seguridad.
   b. Asegúrese de que el grupo de seguridad utilizado por la caché de ElastiCache permita el acceso a los puertos de la caché (6379 y 6380 para clústeres sin servidor y 6379 para clústeres basados en nodos). ElastiCache utiliza estos puertos para aceptar los comandos de Valkey o Redis OSS. Obtenga más información sobre cómo configurar el acceso a los puertos aquí.

Si la conexión sigue siendo difícil, consulte otros pasos en Problemas de las conexiones persistentes.

Errores del cliente de Valkey o Redis OSS

Solo se puede acceder a ElastiCache sin servidor mediante clientes que admitan el protocolo de modo de clúster de Valkey o Redis OSS. Se puede acceder a los clústeres basados en nodos desde los clientes en cualquier modo, según la configuración del clúster.

Si experimenta errores en su cliente, tenga en cuenta lo siguiente:

1. Modo de clúster: si se producen errores CROSSLOT o errores con el comando SELECT, es posible que esté intentando acceder a una caché habilitada con el modo de clúster habilitado con un cliente de Valkey o Redis OSS que no sea compatible con el protocolo del clúster. ElastiCache sin servidor solo es compatible con clientes que admitan el protocolo de clúster de Valkey o Redis OSS. Si desea utilizar Valkey o Redis OSS en el modo de clúster deshabilitado (CMD), debe crear su propio clúster basado en nodos.
2. Errores CROSSLOT: si se produce el error `ERR CROSSLOT Keys in request don't hash to the same slot`, es posible que esté intentando acceder a claves que no pertenecen a la misma ranura de una caché en modo de clúster. Como recordatorio, ElastiCache sin servidor funciona siempre en modo de clúster. Las operaciones con varias claves, las transacciones o los scripts de Lua que contengan varias claves solo están permitidas si todas las claves involucradas se encuentran en la misma ranura de hash.

Para conocer más prácticas recomendadas sobre la configuración de los clientes de Valkey o Redis OSS, consulte esta entrada del blog.

Solución de problemas de alta latencia en ElastiCache sin servidor

Si su carga de trabajo parece experimentar una latencia elevada, puede analizar las métricas `SuccessfulReadRequestLatency` y `SuccessfulWriteRequestLatency` de CloudWatch y comprobar si la latencia está relacionada con ElastiCache sin servidor. Estas métricas miden la latencia interna de ElastiCache sin servidor; no se incluyen la latencia del cliente ni los tiempos de ida y vuelta en la red entre el cliente y el punto de conexión de ElastiCache sin servidor.

Solución de problemas de latencia del cliente

Si observa una latencia elevada en el cliente, pero no un aumento correspondiente en las métricas `SuccessfulReadRequestLatency` y `SuccessfulWriteRequestLatency` de CloudWatch, que miden la latencia del servidor, tenga en cuenta lo siguiente:

- Asegúrese de que el grupo de seguridad permita el acceso a los puertos 6379 y 6380: ElastiCache sin servidor utiliza el puerto 6379 para el punto de conexión principal y el puerto 6380 para el punto de conexión del lector. Algunos clientes establecen la conectividad con ambos puertos para cada nueva conexión, incluso si la aplicación no utiliza la característica Lectura desde réplica. Si su grupo de seguridad no permite el acceso entrante a ambos puertos, el establecimiento de la conexión puede tardar más. Obtenga más información sobre cómo configurar el acceso a los puertos aquí.

Solución de problemas de latencia del servidor

No hay que preocuparse si hay cierta variabilidad y algunos picos ocasionales. No obstante, si la estadística de `Average` muestra un fuerte aumento y este persiste, debe consultar AWS Health Dashboard y el panel de estado personal para obtener más información. Si es necesario, considere la posibilidad de abrir un caso de soporte con Soporte.

Tenga en cuenta las siguientes prácticas recomendadas y estrategias para reducir la latencia:

- Habilitar Lectura desde réplica: si su aplicación lo permite, le recomendamos que habilite la característica Lectura desde réplica en su cliente de Valkey o Redis OSS para escalar las lecturas y lograr una latencia más baja. Cuando está habilitada, ElastiCache sin servidor intenta enrutar las solicitudes de lectura a nodos de caché de réplica que se encuentran en la misma zona de disponibilidad (AZ) que su cliente, con objeto de evitar la latencia de red entre zonas de disponibilidad. Tenga en cuenta que habilitar la característica Lectura desde réplica en su cliente implica que su aplicación acepta, en última instancia, la coherencia de los datos. Es posible que su aplicación reciba datos antiguos durante algún tiempo si intenta leerlos después de escribirlos en una clave.
- Asegúrese de que la aplicación esté implementada en las mismas zonas de disponibilidad que la caché: si la aplicación no se implementa en las mismas zonas de disponibilidad que la caché, es posible que observe una mayor latencia en el cliente. Al crear una caché sin servidor, puede proporcionar las subredes desde las que la aplicación accederá a la caché, y ElastiCache sin servidor crea puntos de conexión de VPC en esas subredes. Asegúrese de que la aplicación esté implementada en las mismas zonas de disponibilidad. De lo contrario, la aplicación podría sufrir un salto entre zonas de disponibilidad al acceder a la caché, lo que provocaría una mayor latencia del cliente.
- Reutilización de conexiones: las solicitudes de ElastiCache sin servidor se realizan mediante una conexión de TCP habilitada para TLS mediante el protocolo RESP. Iniciar la conexión (incluida la autenticación de la conexión, si está configurada) lleva tiempo, por lo que la latencia de la primera solicitud es superior a la típica. Las solicitudes a través de una conexión ya inicializada ofrecen la baja latencia coherente de ElastiCache. Por este motivo, debería considerar la posibilidad de utilizar una agrupación de conexiones o reutilizar las conexiones de Valkey o Redis OSS existentes.
- Velocidad de escalado: ElastiCache sin servidor se escala automáticamente a medida que aumenta la tasa de solicitudes. Un aumento importante y repentino de la tasa de solicitudes, a mayor velocidad a la que se escala ElastiCache sin servidor, puede provocar un aumento de la latencia durante algún tiempo. Por lo general, ElastiCache sin servidor puede aumentar

rápidamente su tasa de solicitudes admitidas y tardar entre 10 y 12 minutos en duplicar la tasa de solicitudes.

- Inspeccione los comandos de ejecución prolongada: algunos comandos de Valkey o Redis OSS, incluidos los scripts de Lua o los comandos de estructuras de datos de gran tamaño, podrían ejecutarse durante mucho tiempo. Para identificar estos comandos, ElastiCache publica métricas de comandos. Con ElastiCache sin servidor, puede usar las métricas `BasedECPUs`.
- Solicitudes con limitación: cuando las solicitudes se limitan en ElastiCache sin servidor, es posible que experimente un aumento de la latencia del cliente en su aplicación. Cuando las solicitudes se limitan en ElastiCache sin servidor, debería ver un aumento en la métrica `ThrottledRequests` de ElastiCache sin servidor. Consulte la siguiente sección para solucionar problemas de las solicitudes con limitaciones.
- Distribución uniforme de claves y solicitudes: en ElastiCache para Valkey y Redis OSS, una distribución desigual de las claves o solicitudes por ranura puede provocar una ranura activa, lo que puede derivarse en una latencia elevada. ElastiCache sin servidor admite hasta 30 000 ECPU/segundo (90 000 ECPU/segundo cuando se utiliza Lectura desde réplica) en una sola ranura, en una carga de trabajo que ejecuta comandos SET/GET sencillos. Le recomendamos que evalúe la distribución de las claves y las solicitudes en las ranuras y que garantice una distribución uniforme si la tasa de solicitudes supera este límite.

Solución de problemas de limitación en ElastiCache sin servidor

En las arquitecturas orientadas a servicios y en los sistemas distribuidos, la limitación de la velocidad a la que los distintos componentes del servicio procesan las llamadas a la API se denomina limitación. Esto suaviza los picos, controla los desajustes en el rendimiento de los componentes y permite recuperaciones más predecibles cuando se produce un evento operativo inesperado. ElastiCache sin servidor está diseñado para este tipo de arquitecturas y la mayoría de clientes de Valkey o Redis OSS tienen reintentos incorporados para las solicitudes con limitación. Cierto grado de limitación no es necesariamente un problema para su aplicación, pero la limitación persistente de una parte sensible a la latencia de su flujo de trabajo de datos puede afectar negativamente a la experiencia del usuario y reducir la eficacia general del sistema.

Cuando las solicitudes se limitan en ElastiCache sin servidor, debería ver un aumento en la métrica `ThrottledRequests` de ElastiCache sin servidor. Si observa un número elevado de solicitudes con limitación, tenga en cuenta lo siguiente:

- Velocidad de escalado: ElastiCache sin servidor se escala automáticamente a medida que ingiere más datos o aumenta la tasa de solicitudes. Si su aplicación se escala a más velocidad que la velocidad a la que se escala ElastiCache sin servidor, es posible que sus solicitudes se limiten mientras que ElastiCache sin servidor se escala para adaptarse a su carga de trabajo. Por lo general, ElastiCache sin servidor puede aumentar el tamaño de almacenamiento rápidamente y tarda entre 10 y 12 minutos en duplicar el tamaño de almacenamiento de la caché.
- Distribución uniforme de claves y solicitudes: en ElastiCache para Valkey y Redis OSS, una distribución desigual de claves o solicitudes por ranura puede generar una ranura activa. Una ranura activa puede hacer que se limiten las solicitudes si la tasa de solicitudes a una única ranura supera las 30 000 ECPU/segundo en una carga de trabajo que ejecute comandos SET/GET sencillos. Del mismo modo, con ElastiCache para Memcached, una tecla de acceso rápido puede hacer que se limiten las solicitudes si la tasa de solicitudes supera las 30 000 ECPU/segundo.
- Lectura desde réplica: si su aplicación lo permite, considere la posibilidad de utilizar la característica Lectura desde réplica. La mayoría de los clientes de Valkey o Redis OSS se pueden configurar para escalar las lecturas con el fin de dirigir las lecturas a los nodos de réplica. Esta característica le permite escalar el tráfico de lectura. Además, ElastiCache sin servidor enruta automáticamente la lectura de las solicitudes de réplica a los nodos de la misma zona de disponibilidad que la aplicación, lo que reduce la latencia. Cuando la característica Lectura desde réplica está habilitada, puede generar hasta 90 000 ECPU/segundo en una sola ranura para cargas de trabajo con comandos SET/GET sencillos.

Problemas de las conexiones persistentes

Se deben verificar los siguientes elementos al solucionar problemas de conectividad persistentes con ElastiCache:

Temas

- Grupos de seguridad
- ACL de red
- Tablas de enrutamiento
- Resolución de los DNS
- Identificación de los problemas con los diagnósticos del lado del servidor
- Validación de la conectividad de red
- Límites relacionados con la red
- Uso de la CPU

- Conexiones que terminan desde el lado del servidor
- Solución de problemas del lado del cliente para instancias de Amazon EC2
- Análisis del tiempo que se tarda en completar una sola solicitud

Grupos de seguridad

Los grupos de seguridad son firewalls virtuales que protegen su cliente ElastiCache (instancia EC2, función de AWS Lambda, contenedor de Amazon ECS, etc.) y la caché de ElastiCache. Además, son firewalls con estado, lo que significa que, después de permitir el tráfico entrante o saliente, las respuestas para ese tráfico se autorizarán automáticamente en el contexto de ese grupo de seguridad específico.

La característica con estado requiere que el grupo de seguridad realice un seguimiento de todas las conexiones autorizadas. Además, hay un límite para las conexiones monitoreadas. Si se alcanza ese límite, las conexiones nuevas producirán errores. Consulte la sección de solución de problemas para obtener ayuda sobre cómo identificar si se han alcanzado los límites en el lado del cliente o en el de ElastiCache.

Puede tener un único grupo de seguridad asignado tanto para el cliente como para el clúster de ElastiCache a la vez o un grupo de seguridad para cada uno de ellos.

En ambos casos, se debe permitir el tráfico de salida del TCP en el puerto ElastiCache desde la fuente y el tráfico de entrada del mismo puerto a ElastiCache. El puerto predeterminado es 11211 para Memcached y 6379 para Valkey o Redis OSS. De forma predeterminada, los grupos de seguridad permiten el tráfico de salida. En este caso, solo se requiere la regla de entrada en el grupo de seguridad de destino.

A fin de obtener más información, consulte Patrones de acceso para obtener la entrada a un clúster de ElastiCache en una Amazon VPC.

ACL de red

Las listas de control de acceso (ACL) de red son reglas sin estado. El tráfico debe estar permitido en ambas direcciones (tanto de entrada como de salida) para tener éxito. Las ACL de red se asignan a subredes, no a recursos específicos. Se puede tener la misma ACL asignada a ElastiCache y al recurso del cliente, sobre todo si se encuentran en la misma subred.

De forma predeterminada, las ACL de red permiten todo el tráfico. Sin embargo, se pueden modificar para denegar o permitir el tráfico. Además, la evaluación de las reglas de las ACL es secuencial,

lo que significa que la regla con el número más bajo que coincida con el tráfico lo permitirá o lo denegará. La configuración mínima para permitir el tráfico de Valkey o Redis OSS es:

ACL de red del lado del cliente:

- Reglas de entrada:
- Número de regla: preferiblemente inferior a cualquier regla de denegación
- Tipo: regla de TCP personalizada
- Protocol: TCP
- Intervalo de puertos: 1024-65535
- Fuente: 0.0.0.0/0 (o establezca reglas individuales para las subredes del clúster de ElastiCache)
- Permitir/Denegar: Permitir

- Reglas salientes
- Número de regla: preferiblemente inferior a cualquier regla de denegación
- Tipo: regla de TCP personalizada
- Protocol: TCP
- Intervalo de puertos: 6379
- Fuente: 0.0.0.0/0 (o las subredes del clúster de ElastiCache. Tenga en cuenta que el uso de las direcciones IP específicas puede causar problemas en caso de que ocurra una conmutación por error o debido a la escalabilidad del clúster)
- Permitir/Denegar: Permitir

ACL de red de ElastiCache:

- Reglas de entrada:
- Número de regla: preferiblemente inferior a cualquier regla de denegación
- Tipo: regla de TCP personalizada
- Protocol: TCP
- Intervalo de puertos: 6379
- Fuente: 0.0.0.0/0 (o establezca reglas individuales para las subredes del clúster de ElastiCache)
- Permitir/Denegar: Permitir

- Reglas salientes
- Número de regla: preferiblemente inferior a cualquier regla de denegación
- Tipo: regla de TCP personalizada
- Protocol: TCP
- Intervalo de puertos: 1024-65535
- Fuente: 0.0.0.0/0 (o las subredes del clúster de ElastiCache. Tenga en cuenta que el uso de las direcciones IP específicas puede causar problemas en caso de que ocurra una conmutación por error o debido a la escalabilidad del clúster)
- Permitir/Denegar: Permitir

Para obtener más información, consulte las ACL de red.

Tablas de enrutamiento

De manera similar a las ACL de red, cada subred puede tener tablas de enrutamiento diferentes. Si los clientes y el clúster de ElastiCache se encuentran en subredes diferentes, asegúrese de que las tablas de enrutamiento permitan la conexión entre ellos.

Los entornos más complejos, los cuales implican varias VPC, enrutamiento dinámico o firewalls de red, pueden dificultar la resolución de los problemas. Consulte Validación de la conectividad de red para confirmar que la configuración de red es adecuada.

Resolución de los DNS

ElastiCache proporciona los puntos de enlace de servicio basados en nombres de dominio (DNS). Los puntos de enlace disponibles son los puntos `Configuration`, `Primary`, `Reader` y `Node`. Para obtener más información, consulte Búsqueda de puntos de enlace de conexión.

En caso de conmutación por error o de modificación del clúster, la dirección asociada al nombre del punto de conexión puede cambiar y se actualizará de forma automática.

Es posible que la configuración de DNS personalizada (es decir, cuando no se emplea el servicio de DNS de la VPC) no conozca los nombres de dominios que proporciona ElastiCache. Asegúrese de que el sistema pueda resolver los puntos de enlace de ElastiCache con éxito mediante herramientas de sistema como `dig` (como se muestra a continuación) o `nslookup`.

```
$ dig +short example.xxxxxx.ng.0001.use1.cache.amazonaws.com
example-001.xxxxxx.0001.use1.cache.amazonaws.com.
```

```
1.2.3.4
```

Además, la resolución de nombres se puede forzar a través del servicio de DNS de la VPC:

```
$ dig +short example.xxxxxx.ng.0001.use1.cache.amazonaws.com @169.254.169.253
example-001.tihewd.0001.use1.cache.amazonaws.com.
1.2.3.4
```

Identificación de los problemas con los diagnósticos del lado del servidor

Las métricas de CloudWatch y la información de tiempo de ejecución del motor de ElastiCache son recursos o datos frecuentes para identificar las posibles causas de los problemas de conexión. En general, un buen análisis comienza con los siguientes elementos:

- Uso de la CPU: Valkey o Redis OSS son aplicaciones de múltiples subprocesos. Sin embargo, la ejecución de cada comando ocurre en un solo subproceso (principal). Por este motivo, ElastiCache proporciona las métricas `CPUUtilization` y `EngineCPUUtilization`. La métrica `EngineCPUUtilization` proporciona el uso de la CPU dedicado al proceso de Valkey o Redis OSS y la `CPUUtilization` el uso en todas las vCPU. Los nodos con más de una vCPU suelen tener valores diferentes para `CPUUtilization` y `EngineCPUUtilization`, el segundo suele ser más alto. Una `EngineCPUUtilization` alta puede producirse por un número elevado de solicitudes u operaciones complejas que toman demasiado tiempo de CPU en completarse. Ambos se pueden identificar por:
  - Un número elevado de solicitudes: verifique si hay aumentos en otras métricas que coincidan con el patrón de `EngineCPUUtilization`. Las métricas útiles son:
    - `CacheHits` y `CacheMisses`: el número de solicitudes correctas o solicitudes que no han encontrado un elemento válido en la caché Si la proporción de los errores en comparación con los aciertos es alta, la aplicación está perdiendo tiempo y consta de recursos con solicitudes poco útiles.
    - `SetTypeCmds` y `GetTypeCmds`: estas métricas, las cuales se encuentran correlacionadas con `EngineCPUUtilization`, pueden ayudar a entender si la carga es mucho más elevada para las solicitudes de escritura (medida por `SetTypeCmds`) o lecturas (medida por `GetTypeCmds`). Si la carga son lecturas en su gran mayoría, el uso de varias réplicas de lectura puede equilibrar las solicitudes en varios nodos y ahorrar las principales para las escrituras. En clústeres con el modo de clúster deshabilitado, el uso de réplicas de lectura se puede realizar mediante la creación de una configuración de conexión adicional en la aplicación mediante el punto de conexión del lector de ElastiCache. Para obtener más

información, consulte Búsqueda de puntos de enlace de conexión. Las operaciones de lectura deben enviarse a esta conexión adicional. Las operaciones de escritura se realizarán a través del punto de conexión principal habitual. En el modo de clúster habilitado, se aconseja utilizar una biblioteca que admita réplicas de lectura por defecto. Con los indicadores correctos, la biblioteca podrá identificar automáticamente la topología del clúster y los nodos de la réplica, así como también habilitar las operaciones de lectura mediante el comando de Valkey o Redis OSS READONLY y enviar las solicitudes de lectura a las réplicas.

- Número elevado de conexiones:
  - `CurrConnections` y `NewConnections`: `CurrConnection` muestra el número de conexiones establecidas en el momento de la recopilación de puntos de datos, mientras que `NewConnections` muestra cuántas conexiones se crearon en el periodo.

    La creación y la gestión de las conexiones implica una sobrecarga significativa de la CPU. Además, el protocolo de establecimiento de comunicación de tres canales del TCP que se necesita para crear conexiones nuevas afectará negativamente a los tiempos de respuesta generales.

    Un nodo de ElastiCache con miles de `NewConnections` por minuto indica que una conexión es creada y utilizada por unos pocos comandos, lo cual no es óptimo. Una práctica recomendada es mantener las conexiones establecidas y reutilizarlas para operaciones nuevas. Esto es posible cuando la aplicación de cliente admite e implementa correctamente la agrupación de conexiones o conexiones persistentes. Con la agrupación de conexiones, el número de `currConnections` no tiene grandes variaciones y `NewConnections` debe ser lo más bajo posible. Valkey y Redis OSS proporcionan un rendimiento óptimo con un pequeño número de currConnections. Mantener la métrica currConnections en el orden de decenas o centenas minimiza el uso de recursos para admitir conexiones individuales tales como los búferes del lado del cliente y los ciclos de la CPU a fin de servir la conexión.
- Rendimiento de la red:
  - Determine la banda ancha: los nodos de ElastiCache tienen una banda ancha de red proporcional al tamaño del nodo. Dado que las aplicaciones tienen características diferentes, los resultados pueden variar según la carga de trabajo. Como, por ejemplo, las aplicaciones que tengan una alta tasa de solicitudes pequeñas tienden a afectar más al uso de la CPU que al rendimiento de la red, mientras que las claves más grandes generarán un mayor uso de la red. Por esta razón, se aconseja probar los nodos con la carga de trabajo real para una mejor comprensión de los límites.

Simular la carga desde la aplicación proporcionaría resultados más precisos. Sin embargo, las herramientas de punto de referencia pueden transmitir una buena noción de los límites.

- Para los casos en los que las solicitudes son principalmente lecturas, el uso de réplicas para operaciones de lectura aliviará la carga en el nodo primario. Si el caso de uso es predominantemente de escrituras, el uso de muchas réplicas amplificará el uso de la red. Por cada byte escrito en el nodo primario, se enviarán N bytes a las réplicas, siendo N el número de réplicas. La práctica recomendada para las cargas de trabajo intensivas de escritura es usar ElastiCache para Redis con el modo de clúster habilitado a fin de que las escrituras se puedan equilibrar en varias particiones o escalar verticalmente hasta un tipo de nodo con más capacidades de red.
- Las métricas de CloudWatch `NetworkBytesIn` y `NetworkBytesOut` proporcionan la cantidad de datos que entran o salen del nodo, respectivamente. `ReplicationBytes` es el tráfico dedicado a la reproducción de datos.

Para obtener más información, consulte Límites relacionados con la red.

- Comandos complejos: los comandos de Redis OSS se ofrecen en un solo subproceso, lo que significa que las solicitudes se ofrecen de forma secuencial. Un solo comando lento puede afectar a otras solicitudes y conexiones, lo que genera tiempos de espera. El uso de comandos que actúan sobre varios valores, claves o tipos de datos debe efectuarse con cuidado. Las conexiones pueden bloquearse o interrumpirse en función del número de parámetros o del tamaño de sus valores de entrada o de salida.

  Un ejemplo notorio es el comando `KEYS`. Analiza todo el espacio de claves en la búsqueda de un patrón dado y bloquea la puesta en marcha de otros comandos durante su ejecución. Redis OSS emplea la notación Big O para describir la complejidad de sus comandos.

  El comando de claves tiene una complejidad de tiempo O(N), siendo N el número de claves en la base de datos. Por lo tanto, cuanto mayor sea el número de claves, más lento será el comando. Sin embargo, el comando `KEYS` puede causar problemas de diferentes maneras. Si no se utiliza un patrón de búsqueda, el comando devolverá todos los nombres de clave disponibles. En las bases de datos con miles o millones de elementos, se creará una enorme salida que saturará a los búferes de red.

  Si se utiliza un patrón de búsqueda, solo las claves que coincidan con el patrón volverán al cliente. No obstante, el motor todavía barrerá todo el espacio de claves en búsqueda de dicho patrón y el tiempo para completar el comando será el mismo.

Una alternativa para KEYS es el comando SCAN. Vuelve a repetir el proceso sobre el espacio de claves y limita las iteraciones en un número específico de elementos, al evitar bloqueos prolongados en el motor.

El escaneo tiene el parámetro COUNT, el cual se utiliza para establecer el tamaño de los bloques de iteración. El valor predeterminado es 10 (10 elementos por iteración).

En función del número de elementos en la base de datos, los bloques de valores COUNT pequeños requerirán más iteraciones para completar un análisis completo, mientras que los valores más grandes mantendrán al motor ocupado durante más tiempo en cada iteración. Mientras que los valores de conteo pequeños harán SCAN más lento en las bases de datos de gran tamaño, los valores más elevados pueden causar los mismos problemas presentados para KEYS.

Por ejemplo, ejecutar el comando SCAN con un valor de conteo en 10 requerirá 100 000 repeticiones en una base de datos con 1 millón de claves. Si el tiempo promedio de ida y vuelta de la red es de 0,5 milisegundos, cerca de 50 000 milisegundos (50 segundos) se utilizarán para transferir solicitudes.

Por otro lado, si el valor de conteo fuera 100 000, se requerirá una sola iteración y solo se gastarían 0,5 ms para transferirla. Sin embargo, el motor se encontraría completamente bloqueado para otras operaciones hasta que el comando termine de analizar todo el espacio de claves.

Además de KEYS, existen otros comandos que son potencialmente dañinos si no se utilizan correctamente. Para ver una lista de todos los comandos, junto con su complejidad de tiempo correspondiente, vaya a Valkey and Redis OSS commands.

Ejemplos de problemas posibles:

- Scripts de Lua: Redis proporciona un intérprete de Lua incrustado, lo que permite la ejecución de scripts en el servidor. Los scripts de Lua en Valkey y Redis OSS se ejecutan en el motor y son atómicos por definición, lo que significa que, mientras se está ejecutando un script, no se permitirá la ejecución de otro comando o script. Los scripts de Lua ofrecen la posibilidad de ejecutar diferentes comandos, algoritmos de toma de decisiones, análisis de datos, entre otros, directamente en el motor. Mientras que la atomicidad de los scripts y la posibilidad de descargar la aplicación son tentadoras, los scripts deben emplearse con cuidado y para pequeñas operaciones. En ElastiCache, el tiempo de ejecución de los scripts de Lua se encuentra limitado a 5 segundos. Los scripts que no se hayan escrito en el espacio de claves

se interrumpirán de manera automática después del periodo de 5 segundos. Para evitar la corrupción de datos y las inconsistencias, el nodo realizará una conmutación por error si la ejecución del script no se ha completado en 5 segundos y ha tenido alguna escritura durante su ejecución. Las transacciones son la alternativa para garantizar la coherencia de varias modificaciones de claves del mismo grupo en Redis OSS. Una transacción permite la ejecución de un bloque de comandos al observar las claves existentes en busca de modificaciones. Si alguna de las claves observadas cambia antes de la finalización de la transacción, se descartan todas las modificaciones.

- Eliminación masiva de elementos: el comando `DEL` acepta varios parámetros, los cuales son los nombres clave que se eliminarán. Las operaciones de eliminación son síncronas y llevarán mucho tiempo de CPU si la lista de parámetros es grande, o si contiene una lista, un conjunto, un conjunto ordenado o un hash grandes (estructuras de datos que contienen varios subelementos). En otras palabras, incluso la eliminación de una sola clave puede tomar un tiempo considerable si tiene muchos elementos. La alternativa a `DEL` es `UNLINK`, que es un comando asíncrono disponible desde Redis OSS 4. Se debe preferir `UNLINK` a `DEL` siempre que sea posible. A partir de ElastiCache para Redis OSS 6x, el parámetro `lazyfree-lazy-user-del` hace que el comando `DEL` se comporte como `UNLINK` cuando se habilita. Para obtener más información, consulte Redis OSS 6.0 Parameter Changes.
- Comandos que actúan sobre varias claves: se mencionó el comando `DEL` como un comando que acepta varios argumentos y su tiempo de ejecución será directamente proporcional a eso. Sin embargo, Redis OSS proporciona muchos más comandos que funcionan de manera similar. Por ejemplo, los comandos `MSET` y `MGET` permiten la inserción o recuperación de varias claves de cadena a la vez. Su uso puede resultar beneficioso para reducir la latencia de la red inherente a varios comandos `SET` o `GET` individuales. Sin embargo, una lista de parámetros extensa afectará al uso de la CPU.

  Aunque el uso de la CPU por sí sola no es la causa de los problemas de conectividad, dedicar demasiado tiempo a procesar uno o varios comandos a través de varias claves puede causar interrupciones en otras solicitudes y aumentar el uso general de la CPU.

  El número y el tamaño de las claves afectarán a la complejidad del comando y, en consecuencia, al tiempo de finalización.

  Otros ejemplos de comandos que pueden actuar sobre varias claves son `HMGET`, `HMSET`, `MSETNX`, `PFCOUNT`, `PFMERGE`, `SDIFF`, `SDIFFSTORE`, `SINTER`, `SINTERSTORE`, `SUNION`, `SUNIONSTORE`, `TOUCH`, `ZDIFF`, `ZDIFFSTORE`, `ZINTER` y `ZINTERSTORE`.

- Comandos que actúan sobre varios tipos de datos: Redis OSS también proporciona comandos que actúan sobre una o varias claves, independientemente de su tipo de datos. ElastiCache para Redis OSS proporciona la métrica `KeyBasedCmds` para supervisar dichos comandos. Esta métrica suma la ejecución de los siguientes comandos en el periodo seleccionado:
  - Complejidad O(N):
    - `KEYS`
  - O(1)
    - `EXISTS`
    - `OBJECT`
    - `PTTL`
    - `RANDOMKEY`
    - `TTL`
    - `TYPE`
    - `EXPIRE`
    - `EXPIREAT`
    - `MOVE`
    - `PERSIST`
    - `PEXPIRE`
    - `PEXPIREAT`
    - `UNLINK (O(N)` para recuperar la memoria. No obstante, la tarea de recuperación de memoria ocurre en un subproceso aparte y no bloquea el motor.
  - Tiempos de complejidad diferentes según el tipo de datos:
    - `DEL`
    - `DUMP`
    - Se estima que el comando `RENAME` tiene una complejidad O(1), pero ejecuta `DEL` internamente. El tiempo de ejecución variará en función del tamaño de la clave que ha sido renombrada.
    - `RENAMENX`
    - `RESTORE`
    - `SORT`

- Hash de gran tamaño: un hash es un tipo de datos que permite una sola clave con varios subelementos de valor de clave. Cada hash puede almacenar 4 294 967 295 elementos y las operaciones en hash grandes pueden volverse costosas. Del mismo modo que `KEYS`, los hashes tienen el comando `HKEYS` con una complejidad de tiempo O(N), siendo N el número de elementos en el hash. Se recomienda emplear `HSCAN` antes que `HKEYS` para evitar comandos de larga ejecución. Los comandos `HDEL`, `HGETALL`, `HMGET`, `HMSET` y `HVALS` se deben utilizar con precaución en hashes grandes.
- Otras estructuras de big data: además de los hashes, existen otras estructuras de datos que pueden ser pesadas para la CPU. Los conjuntos, las listas, los conjuntos ordenados y los Hyperloglogs también pueden demorar en gestionarse en función del tamaño y de los comandos utilizados. Para obtener más información sobre esos comandos, consulte Valkey and Redis OSS commands.

Validación de la conectividad de red

Luego de revisar las configuraciones de red relacionadas con la resolución de DNS, los grupos de seguridad, las ACL de red y las tablas de enrutamiento, la conectividad se puede validar con el VPC Reachability Analyzer y las herramientas del sistema.

Reachability Analyzer probará la conectividad de red y confirmará si se cumplen todos los requisitos y permisos. Para las siguientes pruebas necesitará el ID de ENI (Identificación de interfaz de red elástica) de uno de los nodos de ElastiCache disponibles en la VPC. Para ello, puede realizar lo siguiente:

1. Diríjase a https://console.aws.amazon.com/ec2/v2/home?#NIC:
2. Filtre la lista de interfaces por el nombre del clúster de ElastiCache o por la dirección IP obtenida de las validaciones de DNS.
3. Anote o guarde el ID de ENI. Si se muestran varias interfaces, revise la descripción para confirmar que pertenecen al clúster de ElastiCache correcto y elija una de ellas.
4. Continúe con el siguiente paso.
5. Cree una ruta de análisis en https://console.aws.amazon.com/vpc/home?#ReachabilityAnalyzer y elija las siguientes opciones:

   - Tipo de fuente: elija instance (instancia) si su cliente de ElastiCache se ejecuta en una instancia de Amazon EC2 o Network Interface (Interfaz de red) si utiliza otro servicio, tal

como AWS Fargate Amazon ECS con la red awsvpc, AWS Lambda, etc. y el ID de recurso correspondiente (instancia EC2 o ID de ENI);

- Tipo de destino: elija Network Interface (Interfaz de red) y seleccione la ElastiCache ENI (ENI de ElastiCache) de la lista.
- Puerto de destino: especifique 6379 para ElastiCache para Redis o 11211 para ElastiCache para Memcached. Estos son los puertos definidos con la configuración predeterminada y en este ejemplo se supone que no se modifican.
- Protocolo: TCP

Cree la ruta de análisis y espere unos momentos para obtener el resultado. Si no se puede acceder al estado, abra los detalles del análisis y revise el Explorador de análisis para conocer los detalles en los que se bloquearon las solicitudes.

Si se han superado las pruebas de accesibilidad, proceda a la verificación a nivel del sistema operativo.

Validación de la conectividad de TCP en el puerto de servicios de ElastiCache: en Amazon Linux, `Nping` se encuentra disponible en el paquete `nmap` y puede probar la conectividad de TCP en el puerto de ElastiCache, así como proporcionar el tiempo de ida y vuelta de la red para establecer la conexión. Utilice esta opción para validar la conectividad de red y la latencia actual al clúster de ElastiCache, como se muestra a continuación:

```
$ sudo nping --tcp -p 6379 example.xxxxxx.ng.0001.use1.cache.amazonaws.com

Starting Nping 0.6.40 ( http://nmap.org/nping ) at 2020-12-30 16:48 UTC
SENT (0.0495s) TCP ...
(Output suppressed )

Max rtt: 0.937ms | Min rtt: 0.318ms | Avg rtt: 0.449ms
Raw packets sent: 5 (200B) | Rcvd: 5 (220B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 4.08 seconds
```

De forma predeterminada, `nping` envía 5 sondas con un retraso de 1 segundo entre ellas. Puede utilizar la opción “-c” para aumentar el número de sondas y “-delay” a fin de cambiar el tiempo en que se envía una prueba nueva.

Si las pruebas con el VPC Reachability Analyzer funcionan, pero fracasan con `nping`, pida al administrador del sistema que revise las reglas de firewall basadas en host, las reglas de enrutamiento asimétrico o cualquier otra restricción posible a nivel de sistema operativo.

En la consola de ElastiCache, verifique si el Cifrado en tránsito se encuentra habilitado en los detalles del clúster de ElastiCache. Si el cifrado en tránsito se encuentra habilitado, confirme si la sesión de TLS se puede establecer con el siguiente comando:

```
openssl s_client -connect example.xxxxxx.use1.cache.amazonaws.com:6379
```

Se espera un gran resultado si la conexión y la negociación de TLS son exitosas. Verifique el código de retorno que se encuentra disponible en la última línea, el valor debe ser `0 (ok)`. Si OpenSSL devuelve algo diferente, verifique el motivo del error en https://www.openssl.org/docs/man1.0.2/man1/verify.html#DIAGNOSTICS.

Si se han superado todas las pruebas de infraestructura y de sistema operativo pero la aplicación sigue sin poder conectarse a ElastiCache, verifique si las configuraciones de la aplicación cumplen con la configuración de ElastiCache. Los errores frecuentes son:

- La aplicación no admite el modo clúster de ElastiCache y ElastiCache tiene el modo de clúster habilitado.
- La aplicación no es compatible con TLS/SSL y ElastiCache tiene el cifrado en tránsito habilitado.
- La aplicación es compatible con TLS/SSL, pero no tiene los indicadores de configuración correctos ni las entidades de certificación de confianza.

Límites relacionados con la red

- Número máximo de conexiones: hay límites estrictos para conexiones simultáneas. Cada nodo de ElastiCache permite hasta 65 000 conexiones simultáneas en todos los clientes. Este límite se puede monitorear a través de las métricas `CurrConnections` en CloudWatch. Sin embargo, los clientes también tienen sus límites para las conexiones de salida. En Linux, verifique el rango de puertos efímeros permitido con el comando:

  ```
  # sysctl net.ipv4.ip_local_port_range
  net.ipv4.ip_local_port_range = 32768 60999
  ```

  En el ejemplo anterior, se permitirán 28231 conexiones desde la misma fuente a la misma dirección IP de destino (nodo de ElastiCache) y puerto. El siguiente comando muestra cuántas conexiones existen para un nodo de ElastiCache concreto (IP 1.2.3.4):

```
ss --numeric --tcp state connected "dst 1.2.3.4 and dport == 6379" | grep -vE
 '^State' | wc -l
```

Si el número es demasiado alto, es posible que el sistema se sobrecargue al intentar procesar las solicitudes de conexión. Se recomienda considerar la implementación de técnicas tales como la agrupación de conexiones o conexiones persistentes para controlar las conexiones con mayor facilidad. Siempre que sea posible, configure el grupo de conexiones para limitar el número máximo de conexiones a unos pocos cientos. Además, se recomienda seguir la lógica del retardo para controlar los tiempos de espera u otras excepciones de conexión a fin de evitar la pérdida de conexión en caso de problemas.

- Límites de tráfico de red: verifique las siguientes métricas de CloudWatch para Redis OSS con el objetivo de identificar los posibles límites de la red que ha alcanzado el nodo de ElastiCache:
  - `NetworkBandwidthInAllowanceExceeded`/`NetworkBandwidthOutAllowanceExceeded`: paquetes de red configurados porque el rendimiento superó el límite de banda ancha agregado.

    Es importante tener en cuenta que cada byte escrito en el nodo primario se replicará en N réplicas, siendo N el número de réplicas. Es posible que los clústeres con tipos de nodos pequeños, varias réplicas y solicitudes de escritura intensivas no puedan afrontar al retraso de la reproducción. En estos casos, es una práctica recomendada escalar verticalmente (cambiar el tipo de nodo), escalar horizontalmente (agregar particiones en clústeres en modo de clúster habilitado) y disminuir el número de réplicas o el de escrituras.
  - `NetworkConntrackAllowanceExceeded`: paquetes configurados porque se ha superado el número máximo de conexiones rastreadas en todos los grupos de seguridad asignados al nodo. Es probable que las conexiones nuevas fallen durante este periodo.
  - `NetworkPackets PerSecondAllowanceExceeded`: se ha superado el número máximo de paquetes por segundo. Las cargas de trabajo basadas en una alta tasa de solicitudes muy pequeñas pueden alcanzar este límite antes de la banda ancha máxima.

  Las métricas mencionadas son una manera ideal de confirmar que los nodos alcanzan sus límites de red. No obstante, los límites también son identificables por periodos de estancamiento en las métricas de la red.

  Si dichos periodos se observan durante largos plazos de tiempo, es probable que se produzca un retraso en la reproducción, un aumento de los bytes utilizados para caché y un deterioro en la memoria que se puede liberar, en el intercambio alto y en el uso de la CPU. Las instancias de Amazon EC2 también tienen límites de red a los que se puede realizar un seguimiento mediante

las Métricas de los controladores de ENA. Las instancias de Linux con compatibilidad de red mejorada y los controladores de ENA 2.2.10, o más recientes, pueden controlar los contadores de límites con el comando:

```
# ethtool -S eth0 | grep "allowance_exceeded"
```

Uso de la CPU

La métrica de uso de la CPU es el punto de partida de la investigación y los siguientes elementos pueden ayudar a mermar los posibles problemas en el lado de ElastiCache:

- Registros lentos de Redis OSS: la configuración predeterminada de ElastiCache retiene los últimos 128 comandos que tardaron más de 10 milisegundos en completarse. El historial de comandos lentos se mantiene durante el tiempo de ejecución del motor y se perderá en caso de interrupción o de reinicio. Si la lista alcanza 128 entradas, los eventos antiguos se eliminarán para crear espacio para otros nuevos. El tamaño de la lista de eventos lentos y el tiempo de ejecución que se considera lento puede modificarse a través de los parámetros `slowlog-max-len` y `slowlog-log-slower-than` en un grupo de parámetros personalizados. La lista de registros lentos se puede recuperar al ejecutar `SLOWLOG GET 128` en el motor, siendo 128 los últimos 128 comandos lentos informados. Cada entrada cuenta con los siguientes campos:

```
1) 1) (integer) 1 -----------> Sequential ID
   2) (integer) 1609010767 --> Timestamp (Unix epoch time)of the Event
   3) (integer) 4823378 -----> Time in microseconds to complete the command.
   4) 1) "keys" -------------> Command
      2) "*" ----------------> Arguments
   5) "1.2.3.4:57004"-> Source
```

El evento anterior ocurrió el 26 de diciembre, a las 19:26:07 UTC, tardó 4,8 segundos (4823 ms) en completarse y fue causado por el comando KEYS solicitado desde el cliente 1.2.3.4.

En Linux, la marca de tiempo puede convertirse con la fecha del comando:

```
$ date --date='@1609010767'
Sat Dec 26 19:26:07 UTC 2020
```

Con Python:

```
>>> from datetime import datetime
>>> datetime.fromtimestamp(1609010767)
datetime.datetime(2020, 12, 26, 19, 26, 7)
```

O en Windows con PowerShell:

```
PS D:\Users\user> [datetimeoffset]::FromUnixTimeSeconds('1609010767')
DateTime      : 12/26/2020 7:26:07 PM
UtcDateTime
                : 12/26/2020 7:26:07 PM
LocalDateTime : 12/26/2020 2:26:07 PM
Date          : 12/26/2020 12:00:00 AM
Day           : 26
DayOfWeek
                : Saturday
DayOfYear     : 361
Hour          : 19
Millisecond   : 0
Minute        : 26
Month
                : 12
Offset        : 00:00:00Ticks         : 637446075670000000
UtcTicks
                : 637446075670000000
TimeOfDay     : 19:26:07
Year          : 2020
```

Muchos comandos lentos en un corto periodo de tiempo (el mismo minuto o menos) son motivo de preocupación. Revise la naturaleza de los comandos y cómo se pueden optimizar (consulte los ejemplos anteriores). Si los comandos con complejidad de tiempo O(1) son frecuentes, verifique los demás factores para el elevado uso de la CPU mencionado anteriormente.

- Métricas de latencia: ElastiCache para Redis OSS proporciona métricas de CloudWatch para supervisar la latencia media de diferentes clases de comandos. El punto de datos se calcula al dividir el número total de ejecuciones de comandos en la categoría por el tiempo total de ejecución en el periodo. Es importante entender que los resultados de la métrica de latencia son un agregado de varios comandos. Un solo comando puede provocar resultados inesperados, como tiempos de espera, sin un impacto significativo en las métricas. En tales casos, los eventos de registro lento serían una fuente de información más precisa. La siguiente lista contiene las métricas de latencia disponibles y los comandos respectivos que les afectan.

- EvalBasedCmdsLatency: relacionada con los comandos de lenguaje de scripting Lua, `eval`, `evalsha`;
- GeoSpatialBasedCmdsLatency: `geodist`, `geohash`, `geopos`, `georadius`, `georadiusbymember`, `geoadd`;
- GetTypeCmdsLatency: comandos de lectura, independientemente del tipo de datos;
- HashBasedCmdsLatency: `hexists`, `hget`, `hgetall`, `hkeys`, `hlen`, `hmget`, `hvals`, `hstrlen`, `hdel`, `hincrby`, `hincrbyfloat`, `hmset`, `hset`, `hsetnx`;
- HyperLogLogBasedCmdsLatency: `pfselftest`, `pfcount`, `pfdebug`, `pfadd`, `pfmerge`;
- KeyBasedCmdsLatency: comandos que pueden actuar sobre diferentes tipos de datos: `dump`, `exists`, `keys`, `object`, `pttl`, `randomkey`, `ttl`, `type`, `del`, `expire`, `expireat`, `move`, `persist`, `pexpire`, `pexpireat`, `rename`, `renamenx`, `restoreK`, `sort`, `unlink`;
- ListBasedCmdsLatency: lindex, llen, lrange, blpop, brpop, brpoplpush, linsert, lpop, lpush, lpushx, lrem, lset, ltrim, rpop, rpoplpush, rpush, rpushx;
- PubSubBasedCmdsLatency: psubscribe, publish, pubsub, punsubscribe, subscribe, unsubscribe;
- SetBasedCmdsLatency: `scard`, `sdiff`, `sinter`, `sismember`, `smembers`, `srandmember`, `sunion`, `sadd`, `sdiffstore`, `sinterstore`, `smove`, `spop`, `srem`, `sunionstore`;
- SetTypeCmdsLatency: comandos de escritura, independientemente del tipo de datos;
- SortedSetBasedCmdsLatency: `zcard`, `zcount`, `zrange`, `zrangebyscore`, `zrank`, `zrevrange`, `zrevrangebyscore`, `zrevrank`, `zscore`, `zrangebylex`, `zrevrangebylex`, `zlexcount`, `zadd`. `zincrby`, `zinterstore`, `zrem`, `zremrangebyrank`, `zremrangebyscore`, `zunionstore`, `zremrangebylex`, `zpopmax`, `zpopmin`, `bzpopmin`, `bzpopmax`;
- StringBasedCmdsLatency: `bitcount`, `get`, `getbit`, `getrange`, `mget`, `strlen`, `substr`, `bitpos`, `append`, `bitop`, `bitfield`, `decr`, `decrby`, `getset`, `incr`, `incrby`, `incrbyfloat`, `mset`, `msetnx`, `psetex`, `set`, `setbit`, `setex`, `setnx`, `setrange`;
- StreamBasedCmdsLatency: `xrange`, `xrevrange`, `xlen`, `xread`, `xpending`, `xinfo`, `xadd`, `xgroup`, `readgroup`, `xack`, `xclaim`, `xdel`, `xtrim`, `xsetid`;

- Comandos de tiempo de ejecución de Redis OSS:
  - info commandstats: proporciona una lista de comandos ejecutados desde que se inició el motor, el número de ejecuciones acumuladas, el tiempo total de ejecución y el tiempo promedio de ejecución por comando.

  - client list: proporciona una lista de clientes conectados actualmente e información relevante como el uso de los búferes, el último comando ejecutado, entre otros.
- Copia de seguridad y replicación: las versiones de ElastiCache para Redis OSS anteriores a la 2.8.22 utilizan un proceso bifurcado para crear copias de seguridad y procesar sincronizaciones completas con las réplicas. Este método puede incurrir en una sobrecarga significativa de la memoria para casos de uso intensivos de escritura.

  A partir de ElastiCache Redis OSS 2.8.22, AWS introdujo un método de copia de seguridad y de replicación sin ramificaciones. El método nuevo puede retrasar las escrituras a fin de evitar errores. Ambos métodos pueden causar periodos de mayor uso de la CPU y dar lugar a tiempos de respuesta más altos, lo que, en consecuencia, conlleva a tiempos de espera del cliente durante su ejecución. Siempre verifique si los errores del cliente se producen durante el periodo de copia de seguridad o si la métrica `SaveInProgress` fue 1 en el periodo. Se aconseja programar el periodo de copia de seguridad para periodos de baja utilización con el objetivo de minimizar los posibles problemas con los clientes o los errores de la copia de seguridad.

Conexiones que terminan desde el lado del servidor

La configuración predeterminada de ElastiCache para Redis OSS mantiene las conexiones de cliente establecidas por tiempo indefinido. Sin embargo, en algunos casos, la interrupción de la conexión puede ser deseable. Por ejemplo:

- Los errores en la aplicación cliente pueden hacer que se olviden las conexiones y que se mantengan establecidas con un estado inactivo. Esto se denomina “fuga de conexión”. Su consecuencia es un aumento constante en el número de conexiones establecidas que se observaron en la métrica `CurrConnections`. Este comportamiento puede generar una saturación en el lado cliente o en el de ElastiCache. Cuando no se puede encontrar una solución inmediata desde el lado del cliente, algunos administradores establecen un valor de “tiempo de espera” en su grupo de parámetros de ElastiCache. El tiempo de espera es el tiempo permitido (medido en segundos) para que las conexiones inactivas persistan. Si el cliente no envía una solicitud durante el periodo, el motor terminará la conexión tan pronto como alcance el valor de tiempo de espera. Los valores de tiempo de espera pequeños pueden dar lugar a desconexiones innecesarias de manera que los clientes necesitarán ocuparse correctamente de ellas y volver a conectarse, lo que genera retrasos.
- La memoria empleada para almacenar claves se comparte con los búferes del cliente. Los clientes lentos con grandes solicitudes o respuestas pueden exigir una cantidad significativa de memoria

para operar sus búferes. Las configuraciones predeterminadas de ElastiCache para Redis OSS no restringen el tamaño de los búferes de salida frecuentes del cliente. Si se alcanza el límite de `maxmemory`, el motor intentará expulsar elementos para cumplir con el uso del búfer. En condiciones de memoria extremadamente baja, ElastiCache para Redis OSS puede optar por desconectar clientes que consumen búferes de salida de clientes grandes a fin de liberar memoria y retener el estado del clúster.

Se puede limitar el tamaño de los búferes de cliente mediante configuraciones personalizadas por lo que cuando un cliente alcance ese límite se desconectará. No obstante, los clientes deben ser capaces de resolver desconexiones inesperadas. Los parámetros para manejar el tamaño de los búferes para los clientes regulares son los siguientes:

- client-query-buffer-limit: tamaño máximo de una sola solicitud de entrada;
- client-output-buffer-limit-normal-soft-limit: límite bajo para las conexiones de cliente. La conexión se interrumpirá si permanece por encima del límite bajo durante más tiempo que el tiempo en segundos definido en client-output-buffer-limit-normal-soft-seconds o si alcanza el límite alto;
- client-output-buffer-limit-normal-soft-seconds: tiempo permitido para las conexiones que superan el client-output-buffer-limit-normal-soft-limit;
- client-output-buffer-limit-normal-hard-limit: una conexión que alcance este límite se interrumpirá de inmediato.

Además de los búferes de los clientes frecuentes, las siguientes opciones controlan el búfer para los nodos de réplica y los clientes de publicación/suscripción:

- client-output-buffer-limit-replica-hard-limit;
- client-output-buffer-limit-replica-soft-seconds;
- client-output-buffer-limit-replica-hard-limit;
- client-output-buffer-limit-pubsub-soft-limit;
- client-output-buffer-limit-pubsub-soft-seconds;
- client-output-buffer-limit-pubsub-hard-limit;

Solución de problemas del lado del cliente para instancias de Amazon EC2

La carga y la capacidad de respuesta en el lado del cliente también pueden afectar a las solicitudes a ElastiCache. Los límites de la instancia EC2 y del sistema operativo deben revisarse con cuidado mientras se solucionan problemas de conectividad intermitente o de tiempo de espera. Algunos puntos clave a observar:

- CPU:
  - El uso de la CPU de la instancia EC2: asegúrese de que la CPU no se encuentre saturada o cerca del 100 %. El análisis histórico se puede realizar a través de CloudWatch, sin embargo, tenga en cuenta que la granularidad de los puntos de datos es de 1 minuto (con el monitoreo detallado habilitado) o de 5 minutos.
  - Si utiliza instancias EC2 ampliables, asegúrese de que no se haya agotado el saldo de crédito de la CPU. Esta información se encuentra disponible en la métrica de CloudWatch `CPUCreditBalance`.
  - Los periodos cortos de uso elevado de la CPU pueden generar tiempos de espera que no repercuten en su uso total en CloudWatch. Estos casos requieren un monitoreo en tiempo real con herramientas de sistema operativo como `top`, `ps` y `mpstat`.
- Network
  - Verifique si el rendimiento de la red se encuentra por debajo de los valores aceptables de acuerdo con las capacidades de la instancia. Para obtener más información, consulte Tipos de instancia de Amazon EC2
  - En instancias que dispongan de un controlador de red mejorado `ena`, verifique las estadísticas de ENA para los tiempos de espera o los límites excedidos. Las siguientes estadísticas son útiles para confirmar la saturación de los límites de red:
    - `bw_in_allowance_exceeded`/`bw_out_allowance_exceeded`: número de paquetes moldeados debido a un rendimiento excesivo de entrada o de salida;
    - `conntrack_allowance_exceeded`: número de paquetes descartados debido a los límites de seguimiento de conexiones de los grupos de seguridad. Las conexiones nuevas fallarán cuando este límite se encuentre saturado.
    - `linklocal_allowance_exceeded`: número de paquetes descartados debido a solicitudes excesivas de metadatos de instancia, NTP a través de DNS de la VPC El límite es de 1024 paquetes por segundo para todos los servicios.
    - `pps_allowance_exceeded`: número de paquetes descartados debido a una proporción excesiva de paquetes por segundo. Se puede alcanzar el límite de PPS cuando el tráfico de red consiste en miles o millones de solicitudes muy pequeñas por segundo. El tráfico de ElastiCache se puede optimizar para aprovechar los paquetes de red a través de canalizaciones o comandos que realizan varias operaciones a la vez tal como `MGET` en lugar de `GET`.

Análisis del tiempo que se tarda en completar una sola solicitud

- En la red: `Tcpdump` y `Wireshark` (tshark en la línea de comandos) son herramientas útiles para comprender cuánto tiempo se tomó la solicitud en viajar por la red, en alcanzar al motor de ElastiCache y en regresar. En el próximo ejemplo se resalta una sola solicitud que se creó con el siguiente comando:

```
$ echo ping | nc example.xxxxxx.ng.0001.use1.cache.amazonaws.com 6379
+PONG
```

Del mismo modo que el comando anterior, tcpdump se encontraba en ejecución y volvió:

```
$ sudo tcpdump -i any -nn port 6379 -tt
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
1609428918.917869 IP 172.31.11.142.40966
    > 172.31.11.247.6379: Flags [S], seq 177032944, win 26883, options [mss
 8961,sackOK,TS val 27819440 ecr 0,nop,wscale 7], length 0
1609428918.918071 IP 172.31.11.247.6379 > 172.31.11.142.40966: Flags [S.], seq
 53962565, ack 177032945, win
    28960, options [mss 1460,sackOK,TS val 3788576332 ecr 27819440,nop,wscale 7],
 length 0
1609428918.918091 IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [.], ack 1, win
 211, options [nop,nop,TS val 27819440 ecr 3788576332], length 0
1609428918.918122
    IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [P.], seq 1:6, ack 1, win 211,
 options [nop,nop,TS val 27819440 ecr 3788576332], length 5: RESP "ping"
1609428918.918132 IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [F.], seq 6, ack
    1, win 211, options [nop,nop,TS val 27819440 ecr 3788576332], length 0
1609428918.918240 IP 172.31.11.247.6379 > 172.31.11.142.40966: Flags [.], ack 6, win
 227, options [nop,nop,TS val 3788576332 ecr 27819440], length 0
1609428918.918295
    IP 172.31.11.247.6379 > 172.31.11.142.40966: Flags [P.], seq 1:8, ack 7, win 227,
 options [nop,nop,TS val 3788576332 ecr 27819440], length 7: RESP "PONG"
1609428918.918300 IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [.], ack 8, win
    211, options [nop,nop,TS val 27819441 ecr 3788576332], length 0
1609428918.918302 IP 172.31.11.247.6379 > 172.31.11.142.40966: Flags [F.], seq 8, ack
 7, win 227, options [nop,nop,TS val 3788576332 ecr 27819440], length 0
1609428918.918307
    IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [.], ack 9, win 211, options
 [nop,nop,TS val 27819441 ecr 3788576332], length 0
^C
```

```
10 packets captured
10 packets received by filter
0 packets dropped by kernel
```

De la salida anterior podemos confirmar que el protocolo de establecimiento de comunicación de tres canales de TCP se completó en 222 microsegundos (918091 - 917869) y el comando ping se envió y devolvió en 173 microsegundos (918295 - 918122).

Desde la solicitud hasta la interrupción de la conexión pasaron 438 microsegundos (918307 - 917869). Esos resultados confirmarían que los tiempos de respuesta de la red y del motor son buenos por lo que la investigación puede centrarse en otros componentes.

- En el sistema operativo: `Strace` puede ayudar a identificar brechas de tiempo a nivel de sistema operativo. El análisis de las aplicaciones reales sería mucho más extenso y se aconseja utilizar depuradores o perfiles de aplicaciones especializados. El siguiente ejemplo solo muestra si los componentes del sistema operativo base funcionan como se esperaba, de lo contrario, podría requerirse una investigación adicional. Con el mismo comando `PING` de Redis OSS con `strace` obtenemos:

```
$ echo ping | strace -f -tttt -r -e trace=execve,socket,open,recvfrom,sendto
 nc example.xxxxxx.ng.0001.use1.cache.amazonaws.com (http://
example.xxxxxx.ng.0001.use1.cache.amazonaws.com/)
    6379
1609430221.697712 (+ 0.000000) execve("/usr/bin/nc", ["nc",
 "example.xxxxxx.ng.0001.use"..., "6379"], 0x7fffede7cc38 /* 22 vars */) = 0
1609430221.708955 (+ 0.011231) socket(AF_UNIX, SOCK_STREAM|SOCK_CLOEXEC|
SOCK_NONBLOCK, 0) = 3
1609430221.709084
    (+ 0.000124) socket(AF_UNIX, SOCK_STREAM|SOCK_CLOEXEC|SOCK_NONBLOCK, 0) = 3
1609430221.709258 (+ 0.000173) open("/etc/nsswitch.conf", O_RDONLY|O_CLOEXEC) = 3
1609430221.709637 (+ 0.000378) open("/etc/host.conf", O_RDONLY|O_CLOEXEC) = 3
1609430221.709923
    (+ 0.000286) open("/etc/resolv.conf", O_RDONLY|O_CLOEXEC) = 3
1609430221.711365 (+ 0.001443) open("/etc/hosts", O_RDONLY|O_CLOEXEC) = 3
1609430221.713293 (+ 0.001928) socket(AF_INET, SOCK_DGRAM|SOCK_CLOEXEC|SOCK_NONBLOCK,
 IPPROTO_IP) = 3
1609430221.717419
    (+ 0.004126) recvfrom(3, "\362|
\201\200\0\1\0\2\0\0\0\0\rnotls20201224\6tihew"..., 2048, 0, {sa_family=AF_INET,
 sin_port=htons(53), sin_addr=inet_addr("172.31.0.2")}, [28->16]) = 155
1609430221.717890 (+ 0.000469) recvfrom(3,
 "\204\207\201\200\0\1\0\1\0\0\0\0\rnotls20201224\6tihew"...,
```

```
    65536, 0, {sa_family=AF_INET, sin_port=htons(53),
 sin_addr=inet_addr("172.31.0.2")}, [28->16]) = 139
1609430221.745659 (+ 0.027772) socket(AF_INET, SOCK_STREAM, IPPROTO_TCP) = 3
1609430221.747548 (+ 0.001887) recvfrom(0, 0x7ffcf2f2ca50, 8192,
    0, 0x7ffcf2f2c9d0, [128]) = -1 ENOTSOCK (Socket operation on non-socket)
1609430221.747858 (+ 0.000308) sendto(3, "ping\n", 5, 0, NULL, 0) = 5
1609430221.748048 (+ 0.000188) recvfrom(0, 0x7ffcf2f2ca50, 8192, 0, 0x7ffcf2f2c9d0,
 [128]) = -1 ENOTSOCK
    (Socket operation on non-socket)
1609430221.748330 (+ 0.000282) recvfrom(3, "+PONG\r\n", 8192, 0, 0x7ffcf2f2c9d0,
 [128->0]) = 7
+PONG
1609430221.748543 (+ 0.000213) recvfrom(3, "", 8192, 0, 0x7ffcf2f2c9d0, [128->0]) = 0
1609430221.752110
    (+ 0.003569) +++ exited with 0 +++
```

En el ejemplo anterior, el comando tardó un poco más de 54 milisegundos en completarse (752110 - 697712 = 54398 microsegundos).

Se necesitó una cantidad significativa de tiempo, cerca de 20 ms, para representar a nc y realizar la resolución de nombres (de 697712 a 717890). Luego, se necesitaron 2 ms para crear el socket de TCP (745659 a 747858) y 0,4 ms (747858 a 748330) a fin de enviar y recibir la respuesta para la solicitud.

Temas relacionados

- the section called “Procedimientos recomendados y estrategias de almacenamiento en caché”

Seguridad en Amazon ElastiCache

La seguridad en la nube de AWS es la máxima prioridad. Como cliente de AWS, se beneficia de una arquitectura de red y un centro de datos que se han diseñado para satisfacer los requisitos de seguridad de las organizaciones más exigentes.

La seguridad es una responsabilidad compartida entre AWS y el usuario. El modelo de responsabilidad compartida la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta los servicios de AWS en la nube de AWS. AWS también proporciona servicios que puedes utilizar de forma segura. Auditores independientes prueban y verifican periódicamente la eficacia de nuestra seguridad en el marco de los programas de conformidad de AWS. Para obtener más información sobre los programas de conformidad que se aplican a Amazon ElastiCache, consulte Servicios de AWS en el ámbito del programa de conformidad.
- Seguridad en la nube: su responsabilidad se determina según el servicio de AWSque utiliza. También eres responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y la normativa aplicables.

Este documento ayuda a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza Amazon ElastiCache. En los siguientes temas, se mostrará cómo configurar Amazon ElastiCache para satisfacer los objetivos de seguridad y conformidad. También puede aprender a utilizar otros servicios de AWS que ayudan a monitorear y proteger los recursos de Amazon ElastiCache.

Temas

- Protección de datos en Amazon ElastiCache
- Privacidad del tráfico entre redes
- Identity and Access Management para Amazon ElastiCache
- Validación de la conformidad para Amazon ElastiCache
- Resiliencia en Amazon ElastiCache
- Seguridad de la infraestructura en ElastiCache de AWS
- Actualizaciones de servicios en ElastiCache
- Vulnerabilidades y exposiciones comunes (CVE): vulnerabilidades de seguridad corregidas en ElastiCache

Protección de datos en Amazon ElastiCache

El modelo de responsabilidad compartida de AWS se aplica a la protección de datos de ElastiCache de AWS (ElastiCache). Tal como se describe en este modelo, AWS es responsable de proteger la infraestructura global que ejecuta toda la nube de AWS. Usted es responsable de mantener el control sobre el contenido alojado en esta infraestructura. Este contenido incluye la configuración de seguridad y las tareas de administración de los servicios de AWS que usted utiliza. Para obtener más información sobre la privacidad de datos, consulte las Preguntas frecuentes sobre la privacidad de datos.

Para fines de protección de datos, recomendamos proteger las credenciales de la cuenta de AWS y configurar cuentas individuales con AWS Identity and Access Management (IAM). De esta manera, cada usuario recibe únicamente los permisos necesarios para cumplir con sus obligaciones laborales. También recomendamos proteger sus datos de las siguientes maneras:

- Utiliza la autenticación multifactor (MFA) en cada cuenta.
- Utilice TLS para comunicarse con los recursos de AWS.
- Configure la API y el registro de actividad del usuario con AWS CloudTrail.
- Utilice las soluciones de cifrado de AWS, junto con todos los controles de seguridad predeterminados dentro de los servicios de AWS.
- Utilice avanzados servicios de seguridad administrados, como Amazon Macie, que lo ayuden a detectar y proteger los datos personales almacenados en Amazon S3.

Recomendamos encarecidamente que nunca introduzca información de identificación confidencial, como, por ejemplo, números de cuenta de sus clientes, en los campos de formato libre, como el campo Name (Nombre). Esto incluye cuando se trabaja con ElastiCache u otros servicios de AWS con la consola, la API, la AWS CLI o los SDK de AWS. Es posible que cualquier dato que escriba en ElastiCache o en otros servicios se incluya en los registros de diagnóstico. Cuando proporcione una URL a un servidor externo, no incluya información de credenciales en la URL para validar la solicitud para ese servidor.

Temas

- Seguridad de datos en Amazon ElastiCache

Seguridad de datos en Amazon ElastiCache

Para ayudar a proteger los datos, Amazon ElastiCache y Amazon EC2 disponen de medidas que impiden el acceso no autorizado a los datos del servidor.

Amazon ElastiCache para Memcached también cuenta con características de cifrado para los datos incluidos en cachés que ejecuten las versiones 1.6.12 o posteriores de Memcached.

Amazon ElastiCache con Valkey y Redis OSS cuenta con características de cifrado para los datos en cachés que ejecuten Valkey 7.2 o versiones posteriores, y Redis OSS 3.2.6 (programada para EOL, consulte Programa de fin de vida útil de las versiones de Redis OSS), 4.0.10 o versiones posteriores. Amazon ElastiCache también es compatible con la autenticación de usuarios mediante IAM o AUTH de Valkey y Redis OSS, así como con la autenticación de operaciones de usuarios mediante el control de acceso basado en roles (RBAC).

- El cifrado en tránsito cifra los datos mientras se mueven de un lugar a otro; por ejemplo, entre los nodos del clúster o entre la caché y la aplicación.
- El cifrado en reposo cifra los datos en el disco durante las operaciones de sincronización y copia de seguridad.

ElastiCache permite la autenticación de usuarios mediante IAM y el comando AUTH de Valkey y Redis OSS, así como la autorización de operaciones de usuarios mediante el control de acceso basado en roles (RBAC).

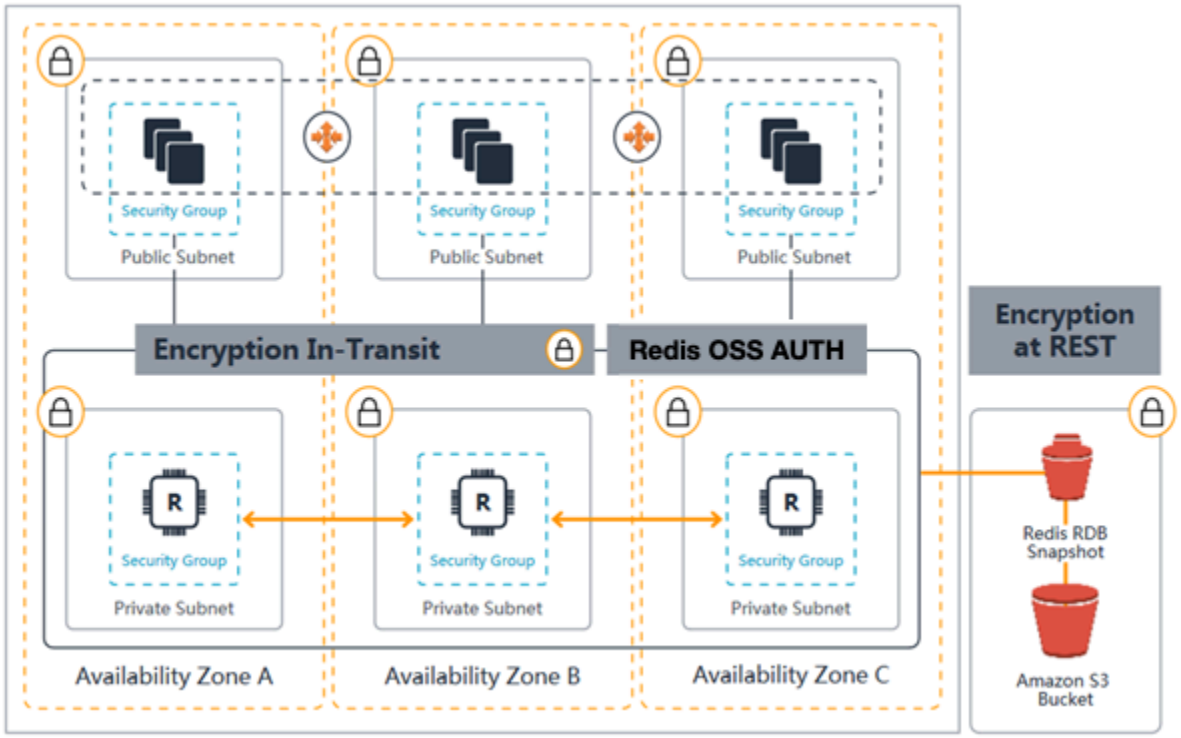


Diagrama de seguridad de ElastiCache para Valkey y Redis OSS

Temas

- Cifrado en tránsito (TLS) de ElastiCache
- Cifrado en reposo en ElastiCache
- Autenticación y autorización

Cifrado en tránsito (TLS) de ElastiCache

Para ayudar a proteger los datos, Amazon ElastiCache y Amazon EC2 disponen de medidas que impiden el acceso no autorizado a los datos del servidor. Al contar con una funcionalidad de cifrado en tránsito, ElastiCache brinda una herramienta que puede utilizar para ayudar a proteger los datos cuando se mueven de una ubicación a otra.

Todas las cachés de Valkey o Redis OSS sin servidor tienen habilitado el cifrado en tránsito. En los clústeres basados en nodos, puede habilitar el cifrado en tránsito en un grupo de replicación estableciendo el parámetro `TransitEncryptionEnabled` en `true` (CLI: `--transit-encryption-enabled`) cuando cree el grupo de replicación. Puede crear el grupo de reproducción a través de la Consola de administración de AWS, la AWS CLI o la API de ElastiCache.

Todas las cachés sin servidor tienen activado el cifrado en tránsito. Para los clústeres basados en nodos, puede habilitar el cifrado en tránsito en un clúster configurando el parámetro `TransitEncryptionEnabled` en `true` (CLI: `--transit-encryption-enabled`) al crear el clúster mediante la operación `CreateCacheCluster` (CLI: `create-cache-cluster`).

Temas

- Información general sobre el cifrado en tránsito
- Condiciones del cifrado en tránsito (Valkey y Redis OSS)
- Condiciones del cifrado en tránsito (Memcached)
- Prácticas recomendadas de cifrado en tránsito
- Más opciones de Valkey y Redis OSS
- Habilitación del cifrado en tránsito para Memcached
- Habilitación del cifrado en tránsito
- Conexión a ElastiCache (Valkey) o Amazon ElastiCache para Redis OSS con cifrado en tránsito mediante valkey-cli
- Activación del cifrado en tránsito en un clúster de Redis OSS basado en nodos con Python
- Procedimientos recomendados para habilitar el cifrado en tránsito
- Conexión a nodos habilitados con el cifrado en tránsito mediante Openssl (Memcached)
- Creación de un cliente TLS Memcached mediante Java
- Creación de un cliente TLS Memcached mediante PHP

Información general sobre el cifrado en tránsito

El cifrado en tránsito de Amazon ElastiCache es una característica que permite reforzar la seguridad de los datos en sus momentos más vulnerables: cuando se trasladan de una ubicación a otra. Como se requiere cierto procesamiento para cifrar y descifrar los datos en los puntos de enlace, habilitar el cifrado en tránsito puede afectar al rendimiento. Debe comparar los datos con y sin cifrado en tránsito para determinar el impacto en el rendimiento de los casos de uso.

El cifrado en tránsito de ElastiCache implementa las siguientes características:

- Conexiones de cliente cifradas: las conexiones de cliente a los nodos de caché están cifradas con TLS.
- Conexiones de servidor cifradas: los datos que se mueven entre los nodos de un clúster están cifrados.

- Autenticación de servidores: los clientes pueden autenticar que se encuentran conectados al servidor correcto.
- Autenticación de clientes: con la característica AUTH de Valkey o Redis OSS, el servidor puede autenticar a los clientes.

Condiciones del cifrado en tránsito (Valkey y Redis OSS)

Cuando planee la implementación de un clúster basado en nodos, debe tener en cuenta las siguientes limitaciones sobre el cifrado en tránsito de Amazon ElastiCache:

- El cifrado en tránsito es compatible con grupos de replicación que ejecuten Valkey 7.2 y versiones posteriores, y Redis OSS versiones 3.2.6, 4.0.10 y posteriores.
- Para un clúster existente, la modificación de la configuración del cifrado en tránsito es compatible con los grupos de replicación que ejecutan Valkey 7.2 y versiones posteriores, y Redis OSS versión 7 y posteriores.
- El cifrado en tránsito solo es compatible con los grupos de reproducción que se ejecutan en una Amazon VPC.
- El cifrado en tránsito no es compatible con los grupos de replicación que ejecutan los siguientes tipos de nodos: M1, M2.

  Para obtener más información, consulte Tipos de nodos compatibles.
- El cifrado en tránsito se habilita estableciendo explícitamente el parámetro `TransitEncryptionEnabled` en `true`.
- El cliente de almacenamiento en caché debe ser compatible con la conectividad TLS y esta debe estar habilitada en la configuración del cliente.
- A partir del 26 de enero de 2026, AWS se actualizará a la versión mínima de TLS admitida que es la 1.2 en ElastiCache para Valkey versión 7.2 y posteriores, y ElastiCache para Redis OSS versión 6 y posteriores. Los clientes deben actualizar su software cliente antes de esa fecha. Esta actualización ayuda a cumplir los requisitos normativos, de seguridad y de conformidad.

Condiciones del cifrado en tránsito (Memcached)

Cuando planee la implementación de un clúster basado en nodos, debe tener en cuenta las siguientes limitaciones sobre el cifrado en tránsito de Amazon ElastiCache:

- El cifrado en tránsito es compatible con los clústeres que ejecutan la versión 1.6.12 y posteriores de Memcached.
- El cifrado en tránsito admite las versiones 1.2 y 1.3 de seguridad de la capa de transporte (TLS).
- El cifrado en tránsito solo es compatible con los clústeres que se ejecutan en una Amazon VPC.
- El cifrado en tránsito no es compatible con los grupos de replicación que ejecutan los siguientes tipos de nodos: M1, M2, M3, R3, T2.

  Para obtener más información, consulte Tipos de nodos compatibles.
- El cifrado en tránsito se habilita estableciendo explícitamente el parámetro `TransitEncryptionEnabled` en `true`.
- Solamente puede habilitar el cifrado en tránsito de un clúster en el momento de su creación. El cifrado en tránsito no se puede habilitar y desactivar modificando un clúster.
- El cliente de almacenamiento en caché debe ser compatible con la conectividad TLS y esta debe estar habilitada en la configuración del cliente.

Prácticas recomendadas de cifrado en tránsito

- Debido al procesamiento requerido para cifrar y descifrar los datos en los puntos de enlace, la implementación del cifrado en tránsito puede reducir el rendimiento. Compare sus datos con y sin cifrado en tránsito para determinar el impacto en el rendimiento de la implementación.
- Puede reducir el impacto en el rendimiento del cifrado en tránsito al mantener las conexiones TSL existentes, ya que la creación de conexiones nuevas puede consumir muchos recursos.

Más opciones de Valkey y Redis OSS

Para obtener más información sobre las opciones disponibles para Valkey y Redis OSS, consulte los siguientes enlaces.

- Cifrado en reposo en ElastiCache
- Autenticación con el comando AUTH de Valkey o Redis OSS
- Control de acceso basado en roles (RBAC)
- Seguridad de ElastiCache y Amazon VPC
- Identity and Access Management para Amazon ElastiCache

Habilitación del cifrado en tránsito para Memcached

Para habilitar el cifrado en tránsito al crear un clúster de Memcached a través de la Consola de administración de AWS, seleccione las siguientes opciones:

- Elija Memcached como motor.
- Elija la versión del motor 1.6.12 o posterior.
- En Encryption in transit (Cifrado en tránsito), elija Enable (Habilitar).

Para ver el procedimiento paso a paso, consulte Creación de un clúster para Valkey o Redis OSS.

Habilitación del cifrado en tránsito

Todas las cachés sin servidor tienen activado el cifrado en tránsito. En un clúster basado en nodos, puede habilitar el cifrado en tránsito mediante la Consola de administración de AWS, la CLI de AWS o la API de ElastiCache.

Habilitación del cifrado en tránsito a través de la Consola de administración de AWS

Habilitación del cifrado en tránsito de un nuevo clúster basado en nodos desde la Consola de administración de AWS

Al diseñar su propia caché, las configuraciones de “desarrollo/prueba” y “producción” con el método “Creación sencilla” tienen el cifrado en tránsito habilitado. Al elegir la configuración, seleccione estas opciones:

- Elija la versión del motor 3.2.6, 4.0.10 o posterior.
- Haga clic en la casilla de verificación situada junto a Habilitar en la opción Cifrado en tránsito.

Para ver el proceso paso por paso, consulte:

- Creación de un clúster de Valkey (modo de clúster deshabilitado) (consola)
- Creación de un clúster de Valkey o Redis OSS (modo de clúster habilitado) (consola)

Habilitación del cifrado en tránsito para un clúster basado en nodos existente desde la Consola de administración de AWS

Habilitar el cifrado en tránsito es un proceso de dos pasos, primero debe configurar el modo de cifrado de tránsito en `preferred`. Este modo permite que los clientes de Valkey o Redis OSS

se conecten mediante conexiones cifradas y no cifradas. Tras migrar todos los clientes de Valkey o Redis OSS para utilizar conexiones cifradas, puede modificar la configuración del clúster para establecer el modo de cifrado de tránsito en `required`. Al configurar el modo de cifrado de tránsito en `required` se eliminarán todas las conexiones no cifradas y solo se permitirán las conexiones cifradas.

Establecimiento de Modo de cifrado en tránsito en Preferido

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. Elija Cachés de Valkey o Cachés de Redis OSS en Recursos de ElastiCache, que se muestra en el panel de navegación de la izquierda.
3. Elija la caché que desea actualizar.
4. Elija Actions (Acciones) y después Modify (Modificar).
5. Elija Enable (Habilitar) en Encryption in transit (Cifrado en tránsito), en la sección Security (Seguridad).
6. Elija Preferred (Preferido) como Transit encryption mode (Modo de cifrado de tránsito).
7. Elija Preview changes (Vista previa de cambios) y guarde los cambios.

Después de migrar todos los clientes de Valkey o Redis OSS para usar conexiones cifradas:

Establecimiento de Modo de cifrado en tránsito en Obligatorio

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. Elija Cachés de Valkey o Cachés de Redis OSS en Recursos de ElastiCache, que se muestra en el panel de navegación de la izquierda.
3. Elija la caché que desea actualizar.
4. Elija Actions (Acciones) y después Modify (Modificar).
5. Elija Required (Obligatorio) como Transit encryption mode (Modo de cifrado de tránsito), en la sección Security (Seguridad).
6. Elija Preview changes (Vista previa de cambios) y guarde los cambios.

Habilitación del cifrado en tránsito a través de la AWS CLI

Para habilitar el cifrado en tránsito al crear un grupo de replicación de Valkey o Redis OSS a través de AWS CLI, utilice el parámetro `transit-encryption-enabled`.

Habilitación del cifrado en tránsito en un nuevo clúster basado en nodos para Valkey o Redis OSS (modo de clúster deshabilitado) (CLI)

Utilice la operación AWS CLI de `create-replication-group` y los parámetros siguientes para crear un grupo de replicación de Valkey o Redis OSS con réplicas que tenga el cifrado en tránsito habilitado:

Parámetros clave:

- **`--engine`**: debe ser `valkey` o `redis`.
- **`--engine-version`**: si el motor es Redis OSS, debe ser la versión 3.2.6, 4.0.10 o posterior.
- **`--transit-encryption-enabled`**: obligatorio. Si habilita el cifrado en tránsito, también deberá proporcionar un valor para el parámetro `--cache-subnet-group`.
- **`--num-cache-clusters`**: debe ser, como mínimo, 1. El valor máximo para este parámetro es seis.

Para obtener más información, consulte los siguientes temas:

- Creación de un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) desde cero (AWS CLI)
- create-replication-group

Habilitación del cifrado en tránsito en un nuevo clúster basado en nodos para Valkey o Redis OSS (modo de clúster habilitado) (CLI)

Utilice la operación `create-replication-group` de la AWS CLI y los siguientes parámetros para crear un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado) que tenga el cifrado en tránsito habilitado:

Parámetros clave:

- **`--engine`**: debe ser `valkey` o `redis`.
- **`--engine-version`**: si el motor es Redis OSS, debe ser la versión 3.2.6, 4.0.10 o posterior.

- **`--transit-encryption-enabled`**: obligatorio. Si habilita el cifrado en tránsito, también deberá proporcionar un valor para el parámetro `--cache-subnet-group`.
- Utilice uno de los siguientes conjuntos de parámetros para especificar la configuración de los grupos de nodos del grupo de reproducción:
  - **`--num-node-groups`**: especifica el número de particiones (grupos de nodos) de este grupo de reproducción. El valor máximo de este parámetro es 500.

    `--replicas-per-node-group`: especifica el número de nodos de réplica de cada grupo de nodos. El valor especificado aquí se aplica a todos los fragmentos de este grupo de reproducción. El valor máximo de este parámetro es 5.
  - **`--node-group-configuration`**: especifica por separado la configuración de cada partición.

Para obtener más información, consulte los siguientes temas:

- Creación de un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado) desde cero (AWS CLI)
- create-replication-group

Habilitación del cifrado en tránsito para un clúster existente con la AWS CLI

Habilitar el cifrado en tránsito es un proceso de dos pasos, primero debe configurar el modo de cifrado de tránsito en `preferred`. Este modo permite que los clientes de Valkey o Redis OSS se conecten mediante conexiones cifradas y no cifradas. Tras migrar todos los clientes de Valkey o Redis OSS para utilizar conexiones cifradas, puede modificar la configuración del clúster para establecer el modo de cifrado de tránsito en `required`. Al configurar el modo de cifrado de tránsito en `required` se eliminarán todas las conexiones no cifradas y solo se permitirán las conexiones cifradas.

Utilice la operación `modify-replication-group` de la AWS CLI y los siguientes parámetros para actualizar un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado) que tenga el cifrado en tránsito deshabilitado.

Para habilitar el cifrado en tránsito

1. Establecer el modo de cifrado en tránsito en `preferred`, mediante los siguientes parámetros
   - **`--transit-encryption-enabled`**: obligatorio.

- **`--transit-encryption-mode`**: debe establecerse en `preferred`.

2. Establecer el modo de cifrado en tránsito en `required`, mediante los siguientes parámetros:

   - **`--transit-encryption-enabled`**: obligatorio.
   - **`--transit-encryption-mode`**: debe establecerse en `required`.

Conexión a ElastiCache (Valkey) o Amazon ElastiCache para Redis OSS con cifrado en tránsito mediante valkey-cli

Para obtener acceso a los datos desde cachés de ElastiCache para Redis OSS habilitadas con el cifrado en tránsito, se utilizan clientes que usan la capa de conexión segura (SSL). También puede utilizar valkey-cli con TLS/SSL en Amazon Linux y Amazon Linux 2. Si su cliente no admite TLS, puede usar el comando `stunnel` en el host de su cliente para crear un túnel de SSL hacia los nodos de Redis OSS.

Conexión cifrada con Linux

Para utilizar valkey-cli para conectarse a un clúster de Valkey o Redis OSS que tiene habilitado el cifrado en tránsito en Amazon Linux 2 o Amazon Linux, siga estos pasos.

1. Descargue y compile la utilidad valkey-cli. Esta utilidad se incluye en la distribución de software de Valkey.
2. En el símbolo del sistema de la instancia de EC2, escriba los comandos correspondientes a la versión de Linux que utilice.

   Amazon Linux 2

   Si utiliza Amazon Linux 2, introduzca lo siguiente:

   ```
   sudo yum -y install openssl-devel gcc
   wget https://github.com/valkey-io/valkey/archive/refs/tags/7.2.6.tar.gz
   tar xvzf valkey-7.2.6.tar.gz
   cd valkey-7.2.6
   make distclean
   make valkey-cli BUILD_TLS=yes
   sudo install -m 755 src/valkey-cli /usr/local/bin/
   ```

   Amazon Linux

Si utiliza Amazon Linux, introduzca lo siguiente:

```
sudo yum install gcc jemalloc-devel openssl-devel tcl tcl-devel clang wget
wget https://github.com/valkey-io/valkey/archive/refs/tags/8.0.0.tar.gz
tar xvzf valkey-8.0.0.tar.gz
cd valkey-8.0.0
make valkey-cli CC=clang BUILD_TLS=yes
sudo install -m 755 src/valkey-cli /usr/local/bin/
```

En Amazon Linux, es posible que también deba ejecutar los siguientes pasos adicionales:

```
sudo yum install clang
CC=clang make
sudo make install
```

3. Cuando haya descargado e instalado la utilidad valkey-cli, recomendamos ejecutar el comando opcional `make-test`.
4. Para conectarse a un clúster con el cifrado y la autenticación habilitados, escriba este comando:

```
valkey-cli -h Primary or Configuration Endpoint --tls -a 'your-password' -p 6379
```

> **Note**
>
> Si instala redis6 en Amazon Linux 2023, ahora puede usar el comando `redis6-cli` en lugar de `valkey-cli`:
>
> ```
> redis6-cli -h Primary or Configuration Endpoint --tls -p 6379
> ```

Conexión cifrada con stunnel

Para utilizar valkey-cli para conectarse a un clúster de Redis OSS habilitado con cifrado en tránsito utilizando stunnel, siga estos pasos.

1. Utilice SSH para conectarse al cliente e instalar `stunnel`.

```
sudo yum install stunnel
```

2. Ejecute el siguiente comando para crear y editar el archivo `'/etc/stunnel/valkey-cli.conf'` de forma simultánea para añadir un punto de conexión del clúster de ElastiCache para Redis OSS a uno o varios parámetros de conexión, mediante la salida que se proporciona a continuación como plantilla.

```
vi /etc/stunnel/valkey-cli.conf


fips = no
setuid = root
setgid = root
pid = /var/run/stunnel.pid
debug = 7
delay = yes
options = NO_SSLv2
options = NO_SSLv3
[valkey-cli]
   client = yes
   accept = 127.0.0.1:6379
   connect = primary.ssltest.wif01h.use1.cache.amazonaws.com:6379
[valkey-cli-replica]
   client = yes
   accept = 127.0.0.1:6380
   connect = ssltest-02.ssltest.wif01h.use1.cache.amazonaws.com:6379
```

   En este ejemplo, el archivo de configuración tiene dos conexiones, `valkey-cli` y `valkey-cli-replica`. Los parámetros se establecen como sigue:

   - client se establece en sí para especificar que esta instancia de stunnel es un cliente.
   - accept se establece en la IP del cliente. En este ejemplo, el principal se establece en el valor predeterminado de Redis OSS 127.0.0.1 en el puerto 6379. La réplica debe llamar a otro puerto y establecerlo en 6380. Puede utilizar los puertos efímeros 1024-65535. Para obtener más información, consulte Puertos efímeros en la Guía del usuario de Amazon VPC.
   - connect se establece en el punto de conexión de servidor de Redis OSS. Para obtener más información, consulte Búsqueda de puntos de conexión en ElastiCache.

3. Se inicia `stunnel`.

```
sudo stunnel /etc/stunnel/valkey-cli.conf
```

Use el comando `netstat` para confirmar que los túneles se han iniciado.

```
sudo netstat -tulnp | grep -i stunnel

tcp        0      0 127.0.0.1:6379              0.0.0.0:*                   LISTEN
     3189/stunnel
tcp        0      0 127.0.0.1:6380              0.0.0.0:*                   LISTEN
     3189/stunnel
```

4. Conéctese al nodo de Redis OSS cifrado mediante el punto de conexión local del túnel.

   - Si no se ha utilizado ninguna contraseña de AUTH al crear el clúster de ElastiCache para Redis OSS, en este ejemplo se utiliza valkey-cli para conectarse al servidor de ElastiCache para Redis OSS utilizando la ruta completa para valkey-cli en Amazon Linux:

     ```
     /home/ec2-user/redis-7.2.5/src/valkey-cli -h localhost -p 6379
     ```

     Si se utilizó la contraseña de AUTH durante la creación del clúster de Redis OSS, en este ejemplo se utiliza valkey-cli para conectarse al servidor de Redis OSS utilizando la ruta completa para valkey-cli, en Amazon Linux:

     ```
      /home/ec2-user/redis-7.2.5/src/valkey-cli -h localhost -p 6379 -a my-secret-
     password
     ```

   O BIEN

   - Cambie el directorio a redis-7.2.5 y proceda del modo siguiente:

     Si no se ha utilizado ninguna contraseña de AUTH al crear el clúster de ElastiCache para Redis OSS, en este ejemplo se utiliza valkey-cli para conectarse al servidor de ElastiCache para Redis OSS utilizando la ruta completa para valkey-cli en Amazon Linux:

     ```
     src/valkey-cli -h localhost -p 6379
     ```

     Si se utilizó la contraseña de AUTH durante la creación del clúster de Redis OSS, en este ejemplo se utiliza valkey-cli para conectarse al servidor de Valkey o Redis OSS utilizando la ruta completa para valkey-cli, en Amazon Linux:

```
src/valkey-cli -h localhost -p 6379 -a my-secret-password
```

En este ejemplo, se utiliza Telnet para conectarse al servidor de Valkey o Redis OSS.

```
telnet localhost 6379

Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
auth MySecretPassword
+OK
get foo
$3
bar
```

5. Para detener y cerrar los túneles SSL, ejecute `pkill` en el proceso de stunnel.

```
sudo pkill stunnel
```

Activación del cifrado en tránsito en un clúster de Redis OSS basado en nodos con Python

La siguiente guía mostrará cómo habilitar el cifrado en tránsito en un clúster de Redis OSS 7.0 que se haya creado originalmente con el cifrado en tránsito deshabilitado. Los clientes de TCP y TLS seguirán comunicándose con el clúster durante este proceso sin tiempo de inactividad.

Boto3 obtendrá las credenciales que necesita (`aws_access_key_id`, `aws_secret_access_key` y `aws_session_token`) de las variables de entorno. Esas credenciales se pegarán previamente en el mismo terminal bash donde ejecutaremos `python3` para procesar el código Python que se muestra en esta guía. El código del ejemplo siguiente se procesó a partir de una instancia de EC2 que se inició en la misma VPC que se utilizará para crear el clúster de ElastiCache Redis OSS en ella.

> **Note**
>
> - En los siguientes ejemplos se utiliza el SDK de Boto3 para las operaciones de administración de ElastiCache (creación de clústeres o usuarios) y redis-py/redis-py-cluster para el manejo de datos.

- Debe utilizar al menos la versión 1.26.39 de boto3 (=~) para utilizar la migración de TLS en línea con la API de modificación del clúster.
- ElastiCache admite la migración de TLS en línea solo para los clústeres con Valkey versión 7.2 y posteriores y Redis OSS versión 7.0 o posteriores. Por lo tanto, si tiene un clúster que ejecuta una versión de Redis OSS anterior a la 7.0, deberá actualizar la versión de Redis OSS del clúster. Para obtener más información sobre las diferencias de las versiones, consulte Diferencias de compatibilidad y comportamiento de las versiones principales del motor con Redis OSS.

Temas

- Definición de las constantes de cadena que iniciarán el clúster de ElastiCache Valkey o Redis OSS
- Definir las clases para la configuración del clúster
- Definir una clase que representará el propio clúster
- (Opcional) Creación de una clase contenedora para demostrar la conexión del cliente al clúster de Valkey o Redis OSS
- Cree la función principal que muestre el proceso de cambio de la configuración de cifrado en tránsito

Definición de las constantes de cadena que iniciarán el clúster de ElastiCache Valkey o Redis OSS

Primero, definamos algunas constantes de cadena de Python simples que contendrán los nombres de las entidades de AWS necesarias para crear el clúster de ElastiCache como `security-group`, `Cache Subnet group` y un `default parameter group`. Todas estas entidades de AWS se deben crear por adelantado en la cuenta de AWS de la región que desee utilizar.

```
#Constants definitions
SECURITY_GROUP = "sg-0492aa0a29c558427"
CLUSTER_DESCRIPTION = "This cluster has been launched as part of the online TLS
 migration user guide"
EC_SUBNET_GROUP = "client-testing"
DEFAULT_PARAMETER_GROUP_REDIS_7_CLUSTER_MODE_ENABLED = "default.redis7.cluster.on"
```

Definir las clases para la configuración del clúster

Ahora, vamos a definir algunas clases sencillas de Python que representarán una configuración de un clúster, que contendrá metadatos sobre el clúster, como la versión de Valkey o Redis OSS, el tipo de instancia y si el cifrado en tránsito (TLS) está habilitado o deshabilitado.

```
#Class definitions

class Config:
    def __init__(
        self,
        instance_type: str = "cache.t4g.small",
        version: str = "7.0",
        multi_az: bool = True,
        TLS: bool = True,
        name: str = None,
    ):
        self.instance_type = instance_type
        self.version = version
        self.multi_az = multi_az
        self.TLS = TLS
        self.name = name or f"tls-test"

    def create_base_launch_request(self):
        return {
            "ReplicationGroupId": self.name,
            "TransitEncryptionEnabled": self.TLS,
            "MultiAZEnabled": self.multi_az,
            "CacheNodeType": self.instance_type,
            "Engine": "redis",
            "EngineVersion": self.version,
            "CacheSubnetGroupName": EC_SUBNET_GROUP ,
            "CacheParameterGroupName":
 DEFAULT_PARAMETER_GROUP_REDIS_7_CLUSTER_MODE_ENABLED ,
            "ReplicationGroupDescription": CLUSTER_DESCRIPTION,
            "SecurityGroupIds": [SECURITY_GROUP],
        }

class ConfigCME(Config):
    def __init__(
        self,
        instance_type: str = "cache.t4g.small",
        version: str = "7.0",
```

```
        multi_az: bool = True,
        TLS: bool = True,
        name: str = None,
        num_shards: int = 2,
        num_replicas_per_shard: int = 1,
    ):
        super().__init__(instance_type, version, multi_az, TLS, name)
        self.num_shards = num_shards
        self.num_replicas_per_shard = num_replicas_per_shard

    def create_launch_request(self) -> dict:
        launch_request = self.create_base_launch_request()
        launch_request["NumNodeGroups"] = self.num_shards
        launch_request["ReplicasPerNodeGroup"] = self.num_replicas_per_shard
        return launch_request
```

Definir una clase que representará el propio clúster

Ahora, vamos a definir algunas clases sencillas de Python que representarán el propio clúster de ElastiCache Valkey o Redis OSS. Esta clase tendrá un campo de cliente que contendrá un cliente de boto3 para las operaciones de administración de ElastiCache, como la creación del clúster y la consulta de la API de ElastiCache.

```
import botocore.config
import boto3

# Create boto3 client
def init_client(region: str = "us-east-1"):
    config = botocore.config.Config(retries={"max_attempts": 10, "mode": "standard"})
    init_request = dict()
    init_request["config"] = config
    init_request["service_name"] = "elasticache"
    init_request["region_name"] = region
    return boto3.client(**init_request)


class ElastiCacheClusterBase:
    def __init__(self, name: str):
        self.name = name
        self.elasticache_client = init_client()

    def get_first_replication_group(self):
        return self.elasticache_client.describe_replication_groups(
```

```
        ReplicationGroupId=self.name
        )["ReplicationGroups"][0]

    def get_status(self) -> str:
        return self.get_first_replication_group()["Status"]

    def get_transit_encryption_enabled(self) -> bool:
        return self.get_first_replication_group()["TransitEncryptionEnabled"]

    def is_available(self) -> bool:
        return self.get_status() == "available"

    def is_modifying(self) -> bool:
        return self.get_status() == "modifying"

    def wait_for_available(self):
        while True:
            if self.is_available():
                break
            else:
                time.sleep(5)

    def wait_for_modifying(self):
        while True:
            if self.is_modifying():
                break
            else:
                time.sleep(5)

    def delete_cluster(self) -> bool:
        self.elasticache_client.delete_replication_group(
            ReplicationGroupId=self.name, RetainPrimaryCluster=False
        )

    def modify_transit_encryption_mode(self, new_transit_encryption_mode: str):
        # generate api call to migrate the cluster to TLS preffered or to TLS required
            self.elasticache_client.modify_replication_group(
                ReplicationGroupId=self.name,
                TransitEncryptionMode=new_transit_encryption_mode,
                TransitEncryptionEnabled=True,
                ApplyImmediately=True,
            )
        self.wait_for_modifying()
```

```
 class ElastiCacheClusterCME(ElastiCacheClusterBase):
    def __init__(self, name: str):
        super().__init__(name)

    @classmethod
    def launch(cls, config: ConfigCME = None) -> ElastiCacheClusterCME:
        config = config or ConfigCME()
        print(config)
        new_cluster = ElastiCacheClusterCME(config.name)
        launch_request = config.create_launch_request()
        new_cluster.elasticache_client.create_replication_group(**launch_request)
        new_cluster.wait_for_available()
        return new_cluster

    def get_configuration_endpoint(self) -> str:
        return self.get_first_replication_group()["ConfigurationEndpoint"]["Address"]

#Since the code can throw exceptions, we define this class to make the code more
 readable and
#so we won't forget to delete the cluster
class ElastiCacheCMEManager:
    def __init__(self, config: ConfigCME = None):
        self.config = config or ConfigCME()

    def __enter__(self) -> ElastiCacheClusterCME:
        self.cluster = ElastiCacheClusterCME.launch(self.config)
        return self.cluster

    def __exit__(self, exc_type, exc_val, exc_tb):
        self.cluster.delete_cluster()
```

(Opcional) Creación de una clase contenedora para demostrar la conexión del cliente al clúster de Valkey o Redis OSS

Ahora, vamos a crear una clase contenedora para el cliente de `redis-py-cluster`. Esta clase contenedora permitirá rellenar previamente el clúster con algunas claves y luego ejecutar comandos `get` repetidos aleatorios.

> **Note**
>
> Este es un paso opcional, pero simplifica el código de la función principal que viene en un paso posterior.

```
import redis
improt random
from time import perf_counter_ns, time


class DowntimeTestClient:
    def __init__(self, client):
        self.client = client

        # num of keys prefilled
        self.prefilled = 0
        # percent of get above prefilled
        self.percent_get_above_prefilled = 10 # nil result expected when get hit above
 prefilled
        # total downtime in nano seconds
        self.downtime_ns = 0
        # num of success and fail operations
        self.success_ops = 0
        self.fail_ops = 0
        self.connection_errors = 0
        self.timeout_errors = 0


    def replace_client(self, client):
        self.client = client

    def prefill_data(self, timelimit_sec=60):
        end_time = time() + timelimit_sec
        while time() < end_time:
            self.client.set(self.prefilled, self.prefilled)
            self.prefilled += 1

    # unsuccesful operations throw exceptions
    def _exec(self, func):
        try:
            start_ns = perf_counter_ns()
            func()
            self.success_ops += 1
            elapsed_ms = (perf_counter_ns() - start_ns) // 10 ** 6
            # upon succesful execution of func
            # reset random_key to None so that the next command
            # will use a new random key
            self.random_key = None
```

```
        except Exception as e:
            elapsed_ns = perf_counter_ns() - start_ns
            self.downtime_ns += elapsed_ns
            # in case of failure- increment the relevant counters so that we will keep
track
            # of how many connection issues we had while trying to communicate with
            # the cluster.
            self.fail_ops += 1
            if e.__class__ is redis.exceptions.ConnectionError:
                self.connection_errors += 1
            if e.__class__ is redis.exceptions.TimeoutError:
                self.timeout_errors += 1

    def _repeat_exec(self, func, seconds):
        end_time = time() + seconds
        while time() < end_time:
            self._exec(func)

    def _new_random_key_if_needed(self, percent_above_prefilled):
        if self.random_key is None:
            max = int((self.prefilled * (100 + percent_above_prefilled)) / 100)
            return random.randint(0, max)
        return self.random_key

    def _random_get(self):
        key = self._new_random_key_if_needed(self.percent_get_above_prefilled)
        result = self.client.get(key)
        # we know the key was set for sure only in the case key < self.prefilled
        if key < self.prefilled:
            assert result.decode("UTF-8") == str(key)


    def repeat_get(self, seconds=60):
        self._repeat_exec(self._random_get, seconds)

    def get_downtime_ms(self) -> int:
        return self.downtime_ns // 10 ** 6


    def do_get_until(self, cond_check):
        while not cond_check():
            self.repeat_get()
        # do one more get cycle once condition is met
```

```
        self.repeat_get()
```

Cree la función principal que muestre el proceso de cambio de la configuración de cifrado en tránsito

Ahora, definamos la función principal, que hará lo siguiente:

1. Cree el clúster con el cliente de boto3 ElastiCache.
2. Inicialice el cliente de `redis-py-cluster` que se conectará al clúster con una conexión TCP clara sin TLS.
3. El cliente de `redis-py-cluster` rellena previamente el clúster con algunos datos.
4. El cliente de boto3 activará la migración de TLS de sin TLS a TLS preferido.
5. Mientras se migra el clúster a TLS `Preferred`, el cliente de TCP de `redis-py-cluster` enviará operaciones `get` repetidas al clúster hasta que finalice la migración.
6. Una vez finalizada la migración a TLS `Preferred`, confirmaremos que el clúster admite el cifrado en tránsito. Después, crearemos un cliente de `redis-py-cluster` que se conectará al clúster con TLS.
7. Enviaremos algunos comandos `get` utilizando el nuevo cliente de TLS y el antiguo cliente TCP.
8. El cliente de boto3 activará la migración de TLS de TLS `Preferred` a TLS requerido.
9. Mientras se migra el clúster a TLS requerido, el cliente de TLS de redis-py-clúster enviará operaciones `get` repetidas al clúster hasta que finalice la migración.

```
import redis

def init_cluster_client(
    cluster: ElastiCacheClusterCME, prefill_data: bool, TLS: bool = True) ->
 DowntimeTestClient:
    # we must use for the host name the cluster configuration endpoint.
    redis_client = redis.RedisCluster(
        host=cluster.get_configuration_endpoint(), ssl=TLS, socket_timeout=0.25,
 socket_connect_timeout=0.1
    )
    test_client = DowntimeTestClient(redis_client)
    if prefill_data:
        test_client.prefill_data()
    return test_client

if __name__ == '__main__':
    config = ConfigCME(TLS=False, instance_type="cache.m5.large")
```

```
   with ElastiCacheCMEManager(config) as cluster:
       # create a client that will connect to the cluster with clear tcp connection
       test_client_tcp = init_cluster_client(cluster, prefill_data=True, TLS=False)

      # migrate the cluster to TLS Preferred
       cluster.modify_transit_encryption_mode(new_transit_encryption_mode="preferred")

       # do repeated get commands until the cluster finishes the migration to TLS
Preferred
       test_client_tcp.do_get_until(cluster.is_available)

      # verify that in transit encryption is enabled so that clients will be able to
connect to the cluster with TLS
       assert cluster.get_transit_encryption_enabled() == True

      # create a client that will connect to the cluster with TLS connection.
       # we must first make sure that the cluster indeed supports TLS
       test_client_tls = init_cluster_client(cluster, prefill_data=True, TLS=True)

       # by doing get commands with the tcp client for 60 more seconds
      # we can verify that the existing tcp connection to the cluster still works
       test_client_tcp.repeat_get(seconds=60)

       # do get commands with the new TLS client for 60 more seconds
       test_client_tcp.repeat_get(seconds=60)

      # migrate the cluster to TLS required
       cluster.modify_transit_encryption_mode(new_transit_encryption_mode="required")

      # from this point the tcp clients will be disconnected and we must not use them
anymore.
      # do get commands with the TLS client until the cluster finishes migartion to
TLS required mode.
       test_client_tls.do_get_until(cluster.is_available)
```

Procedimientos recomendados para habilitar el cifrado en tránsito

Antes de habilitar el cifrado en tránsito: asegúrese de gestionar correctamente los registros de DNS

> **Note**
>
> Durante este proceso, cambiaremos y eliminaremos los puntos de conexión antiguos. El uso incorrecto de los puntos de conexión puede provocar que el cliente de Valkey o Redis OSS utilice puntos de conexión antiguos y eliminados, lo que impedirá que se conecte al clúster.

Mientras se migra el clúster de sin TLS a TLS preferido, se conserva el registro de DNS antiguo del punto de conexión de clúster y los nuevos registros de DNS del punto de conexión de configuración del clúster se generan en un formato diferente. Los clústeres activados para TLS utilizan un formato de registros de DNS diferente al de los clústeres no activados para TLS. ElastiCache conservará ambos registros de DNS cuando un clúster esté configurado en `encryption mode: Preferred` de manera que las aplicaciones y otros clientes de Valkey o Redis OSS puedan cambiar de uno a otro. Durante el proceso de migración de TLS se producen los siguientes cambios en los registros de DNS:

Descripción de los cambios en los registros de DNS que se producen al habilitar el cifrado en tránsito

Para clústeres de CME

Cuando un clúster está configurado en “modo de cifrado de tránsito: preferido”:

- Los puntos de conexión de configuración del clúster original para el clúster no activado para TLS permanecerán activos. No habrá tiempo de inactividad cuando el clúster se vuelva a configurar del modo de cifrado TLS “ninguno” a “preferido”.
- Se generarán nuevos puntos de conexión de TLS de Valkey o Redis OSS cuando el clúster se establezca en el modo preferido de TLS. Estos nuevos puntos de conexión se resolverán con las mismas IP que los anteriores (no TLS).
- El nuevo punto de conexión de configuración de TLS de Valkey o Redis OSS se mostrará en la consola de ElastiCache y en la respuesta a la API `describe-replication-group`.

Cuando un clúster está configurado en “modo de cifrado de tránsito: requerido”:

- Se eliminarán los puntos de conexión antiguos que no estén habilitados para TLS. No habrá tiempo de inactividad en los puntos de conexión del clúster de TLS.

- Puede recuperar un nuevo `cluster-configuration-endpoint` desde la consola de ElastiCache o desde la API `describe-replication-group`.

Para clústeres de CMD con la conmutación por error automática habilitada o la conmutación por error automática desactivada

Cuando el grupo de replicación está establecido en “modo de cifrado de tránsito: preferido”:

- El punto de conexión principal y el punto de conexión del lector originales del clúster que no admite TLS permanecerán activos.
- Se generarán nuevos puntos de conexión principales y del lector de TLS cuando el clúster se establezca en el modo `Preferred` de TLS. Estos nuevos puntos de conexión se resolverán con las mismas IP que los anteriores (sin TLS).
- El punto de conexión principal y el punto de conexión del lector nuevos se mostrarán en la consola de ElastiCache y en la respuesta a la API `describe-replication-group`.

Cuando el grupo de replicación está establecido en “modo de cifrado de tránsito: requerido”:

- Se eliminarán los puntos de conexión principales y de lector antiguos que no sean de TLS. No habrá tiempo de inactividad en los puntos de conexión del clúster de TLS.
- Puede recuperar nuevos puntos de conexión principales y del lector desde la consola de ElastiCache o desde la API `describe-replication-group`.

El uso sugerido de los registros de DNS

Para clústeres de CME

- Utilice el punto de conexión de configuración del clúster en lugar de los registros de DNS por nodo en el código de la aplicación. No se recomienda utilizar nombres de DNS por nodo directamente, ya que es posible que cambien y el código de la aplicación interrumpa la conexión con el clúster.
- No codifique el punto de conexión de configuración de un clúster en la aplicación, ya que cambiará durante este proceso.
- No se recomienda tener el punto de conexión de configuración del clúster codificado en la aplicación, ya que se puede cambiar durante este proceso. Una vez completado el cifrado en tránsito, consulte el punto de conexión de configuración del clúster con la API `describe-`

`replication-group` (tal como se ha demostrado anteriormente [en negrita]) y utilice el DNS que obtendrá como respuesta a partir de ese momento.

Para clústeres de CMD con la conmutación por error automática habilitada

- Utilice el punto de conexión principal y el punto de conexión del lector en lugar de los nombres de DNS por nodo del código de la aplicación, ya que los nombres de DNS antiguos por nodo se eliminan y se generan otros nuevos al migrar el clúster de sin TLS a TLS preferido. No se recomienda utilizar nombres de DNS por nodo directamente, ya que es posible que agregue réplicas al clúster en el futuro. Además, cuando la conmutación por error automática está habilitada, el servicio de ElastiCache cambia automáticamente los roles del clúster principal y las réplicas. Se recomienda utilizar el punto de conexión principal y el punto de conexión del lector para ayudarlo a realizar un seguimiento de esos cambios. Por último, utilizar el punto de conexión del lector le ayudará a distribuir las lecturas de las réplicas de manera equitativa entre las réplicas del clúster.
- Tener el punto de conexión principal y el punto de conexión del lector codificados en la aplicación no es conveniente, ya que se pueden cambiar durante el proceso de migración de TLS. Una vez finalizado el cambio de migración a TLS preferido, consulte el punto de conexión principal y el punto de conexión del lector con la API describe-replication-group y utilice el DNS que obtendrá como respuesta a partir de ahora. De esta forma, podrá realizar un seguimiento de los cambios en los puntos de conexión de forma dinámica.

Para clústeres de CMD con la conmutación por error automática desactivada

- Utilice el punto de conexión principal y el punto de conexión del lector en lugar de los nombres de DNS por nodo del código de la aplicación. Cuando la conmutación por error automática está desactivada, usted se encarga de escalar, aplicar parches, conmutación por error y otros procedimientos que el servicio de ElastiCache administra automáticamente cuando la conmutación por error automática está habilitada. Esto le facilita la realización manual del seguimiento de los distintos puntos de conexión. Dado que los nombres de DNS antiguos por nodo se eliminan y se generan otros nuevos al migrar el clúster de sin TLS a TLS preferido, no utilice los nombres de DNS por nodo directamente. Esto es obligatorio para que los clientes puedan conectarse al clúster durante la migración de TLS. Además, se beneficiará de distribuir las lecturas de manera uniforme entre las réplicas cuando utilice el punto de conexión del lector y de llevar un registro de los registros de DNS al agregar o eliminar réplicas del clúster.

- Tener el punto de conexión de configuración del clúster codificado en la aplicación no es conveniente, ya que se puede cambiar durante el proceso de migración de TLS.

Durante el cifrado en tránsito: preste atención a cuándo finaliza el proceso de migración

El cambio del modo de cifrado de tránsito no es inmediato y puede llevar algún tiempo. Esto es especialmente cierto en el caso de los clústeres grandes. Solo cuando el clúster finalice la migración a TLS preferido podrá aceptar y ofrecer conexiones de TCP y TLS. Por lo tanto, no debe crear clientes que intenten establecer conexiones de TLS con el clúster hasta que se complete el cifrado en tránsito.

Hay varias formas de recibir notificaciones cuando el cifrado en tránsito se completa correctamente o no funciona: (No se muestra en el ejemplo de código anterior):

- Uso del servicio de SNS para recibir una notificación cuando se complete el cifrado
- Uso de la API `describe-events` que emitirá un evento cuando se complete el cifrado
- Ver un mensaje en la consola de ElastiCache que indica que se ha completado el cifrado

También puede implementar la lógica en la aplicación para saber si se ha completado el cifrado. En el ejemplo anterior, vimos varias formas de garantizar que el clúster finalice la migración:

- Esperar a que comience el proceso de migración (el estado del clúster cambia a “modificándose”) y esperar a que finalice la modificación (el estado del clúster vuelve a ser “disponible”)
- Afirmar que el clúster ha establecido `transit_encryption_enabled` en Verdadero mediante una consulta a la API `describe-replication-group`.

Después de habilitar el cifrado en tránsito: asegúrese de que los clientes que utiliza estén configurados correctamente

Mientras el clúster esté en el modo TLS preferido, la aplicación debe abrir conexiones de TLS al clúster y utilizar solo esas conexiones. De esta forma, la aplicación no sufrirá tiempo de inactividad al habilitar el cifrado en tránsito. Puede asegurarse de que no haya conexiones de TCP más claras al motor de Valkey o Redis OSS mediante el comando info en la sección de SSL.

```
# SSL
ssl_enabled:yes
ssl_current_certificate_not_before_date:Mar 20 23:27:07 2017 GMT
```

```
ssl_current_certificate_not_after_date:Feb 24 23:27:07 2117 GMT
ssl_current_certificate_serial:D8C7DEA91E684163
tls_mode_connected_tcp_clients:0   (should be zero)
tls_mode_connected_tls_clients:100
```

Conexión a nodos habilitados con el cifrado en tránsito mediante Openssl (Memcached)

Para obtener acceso a los datos desde nodos de ElastiCache para Memcached habilitados con el cifrado en tránsito, se utilizan clientes que utilizan la capa de conexión segura (SSL). También puede utilizar Openssl s_client en Amazon Linux y Amazon Linux 2.

Para utilizar Openssl s_client para conectarse a un clúster de Memcached que tiene habilitado el cifrado en tránsito en Amazon Linux 2 o Amazon Linux:

```
/usr/bin/openssl s_client -connect memcached-node-endpoint:memcached-port
```

Creación de un cliente TLS Memcached mediante Java

Para crear un cliente en modo TLS, haga lo siguiente para inicializar el cliente con el SSLContext apropiado:

```
import java.security.KeyStore;
import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManagerFactory;
import net.spy.memcached.AddrUtil;
import net.spy.memcached.ConnectionFactoryBuilder;
import net.spy.memcached.MemcachedClient;
public class TLSDemo {
    public static void main(String[] args) throws Exception {
        ConnectionFactoryBuilder connectionFactoryBuilder = new
 ConnectionFactoryBuilder();
        // Build SSLContext
        TrustManagerFactory tmf =
 TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm());
        tmf.init((KeyStore) null);
        SSLContext sslContext = SSLContext.getInstance("TLS");
        sslContext.init(null, tmf.getTrustManagers(), null);
        // Create the client in TLS mode
        connectionFactoryBuilder.setSSLContext(sslContext);
        MemcachedClient client = new MemcachedClient(connectionFactoryBuilder.build(),
 AddrUtil.getAddresses("mycluster.fnjyzo.cfg.use1.cache.amazonaws.com:11211"));
```

```
        // Store a data item for an hour.
        client.set("theKey", 3600, "This is the data value");
    }
}
```

Creación de un cliente TLS Memcached mediante PHP

Para crear un cliente en modo TLS, haga lo siguiente para inicializar el cliente con el SSLContext apropiado:

```
<?php

/**
 * Sample PHP code to show how to create a TLS Memcached client. In this example we
 * will use the Amazon ElastiCache Auto Descovery feature, but TLS can also be
 * used with a Static mode client.
 * See Using the ElastiCache Cluster Client for PHP (https://docs.aws.amazon.com/
AmazonElastiCache/latest/dg/AutoDiscovery.Using.ModifyApp.PHP.html) for more
 information
 * about Auto Discovery and persistent-id.
 */

/* Configuration endpoint to use to initialize memcached client.
 * this is only an example */
$server_endpoint = "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";

/* Port for connecting to the cluster.
 * This is only an example     */
$server_port = 11211;

/* Initialize a persistent Memcached client and configure it with the Dynamic client
 mode  */
$tls_client =  new Memcached('persistent-id');
$tls_client->setOption(Memcached::OPT_CLIENT_MODE, Memcached::DYNAMIC_CLIENT_MODE);

/* Add the memcached's cluster server/s */
$tls_client->addServer($server_endpoint, $server_port);

/* Configure the client to use TLS */
if(!$tls_client->setOption(Memcached::OPT_USE_TLS, 1)) {
    echo $tls_client->getLastErrorMessage(), "\n";
    exit(1);
}
```

```
/* Set your TLS context configurations values.
 * See MemcachedTLSContextConfig in memcached-api.php for all configurations */
$tls_config = new MemcachedTLSContextConfig();
$tls_config->hostname = '*.mycluster.fnjyzo.use1.cache.amazonaws.com';
$tls_config->skip_cert_verify = false;
$tls_config->skip_hostname_verify = false;

/* Use the created TLS context configuration object to create OpenSSL's SSL_CTX and set
 it to your client.
 * Note:  These TLS context configurations will be applied to all the servers connected
 to this client. */
$tls_client->createAndSetTLSContext((array)$tls_config);

/* test the TLS connection with set-get scenario: */

 /* store the data for 60 seconds in the cluster.
 * The client will decide which cache host will store this item.
 */
if($tls_client->set('key', 'value', 60)) {
    print "Successfully stored key\n";
} else {
    echo "Failed to set key: ", $tls_client->getLastErrorMessage(), "\n";
    exit(1);
}

/* retrieve the key */
if ($tls_client->get('key') === 'value') {
    print "Successfully retrieved key\n";
} else {
    echo "Failed to get key: ", $tls_client->getLastErrorMessage(), "\n";
    exit(1);
}
```

Para obtener más información sobre el uso del cliente PHP, consulte Instalación de ElastiCache Cluster Client para .PHP.

Cifrado en reposo en ElastiCache

Para ayudar a mantener los datos protegidos, Amazon ElastiCache y Amazon S3 cuentan con diferentes formas que permiten restringir el acceso a los datos de la caché. Para obtener más información, consulte Seguridad de ElastiCache y Amazon VPC y Identity and Access Management para Amazon ElastiCache.

El cifrado en reposo de ElastiCache es una característica que permite aumentar la seguridad de la información al cifrar los datos del disco. Siempre está habilitada en una caché sin servidor. Si está habilitada, cifra los siguientes elementos:

- Disco durante las operaciones de sincronización, copia de seguridad o intercambio
- Copias de seguridad almacenadas en Amazon S3

Los datos almacenados en los SSD (unidades de estado sólido) en clústeres habilitados para la organización de datos en niveles siempre se cifran.

ElastiCache ofrece cifrado en reposo predeterminado (servicio administrado), así como capacidad para utilizar las propias claves de AWS KMS simétricas administradas por el cliente en AWS Key Management Service (KMS). Cuando se haga una copia de seguridad de la caché, elija, en las opciones de cifrado, si desea usar la clave de cifrado predeterminada o una clave administrada por el cliente. Para obtener más información, consulte Activación del cifrado en reposo.

> **Important**
>
> Habilitar el cifrado en reposo en un clúster de Valkey o Redis OSS basado en nodos existente implica eliminar el grupo de replicación existente, después de ejecutar la copia de seguridad y restaurarla en el grupo de replicación.

El cifrado en reposo solo se puede habilitar en una caché en el momento de su creación. Como se requiere cierto procesamiento para cifrar y descifrar los datos, habilitar el cifrado en reposo durante estas operaciones puede afectar al rendimiento. Debe comparar los datos con y sin cifrado en reposo para determinar el impacto en el rendimiento de los casos de uso.

Temas

- Condiciones del cifrado en reposo
- Uso de claves administradas por el cliente desde AWS KMS

- Activación del cifrado en reposo
- Véase también

Condiciones del cifrado en reposo

Cuando planee la implementación del cifrado en reposo de ElastiCache, debe tener en cuenta las siguientes limitaciones del cifrado en reposo de ElastiCache:

- El cifrado en reposo es compatible con los grupos de replicación que ejecutan Valkey 7.2 y versiones posteriores y Redis OSS 4.0.10 o posteriores (para la versión 3.2.6 programada para EOL, consulte Programa de fin de vida útil de las versiones de Redis OSS).
- El cifrado en reposo solo es compatible con los grupos de reproducción que se ejecutan en una Amazon VPC.
- El cifrado en reposo solo se admite en los grupos de reproducción que ejecutan los siguientes tipos de nodos.
  - R7g, R6gd, R6g, R5, R4 y R3
  - M7g, M6g, M5, M4 y M3
  - T4g, T3, T2
  - C7gn

  Para obtener más información, consulte Tipos de nodos compatibles
- El cifrado en reposo se habilita estableciendo explícitamente el parámetro `AtRestEncryptionEnabled` en `true`.
- El cifrado en reposo solamente se puede habilitar en un grupo de reproducción en el momento de su creación. Por tanto, el cifrado en reposo no se puede habilitar y desactivar modificando un grupo de reproducción. Para obtener información acerca de la implementación del cifrado en reposo en un grupo de reproducción existente, consulte Activación del cifrado en reposo.
- Si un clúster utiliza un tipo de nodo de la familia r6gd, los datos almacenados en SSD se cifran independientemente de si el cifrado en reposo está habilitado o no.
- Además, la opción de utilizar la clave administrada por el cliente para el cifrado en reposo no se encuentra disponible en las regiones de AWS GovCloud (us-gov-east-1 y us-gov-west-1).
- Si un clúster utiliza un tipo de nodo de la familia r6gd, los datos almacenados en SSD se cifran con la clave KMS de AWS elegida y administrada por el cliente (o cifrado administrado por servicios en Regiones de GovCloud de AWS).

- Con Memcached, el cifrado en reposo solo se admite en las cachés sin servidor.
- Al utilizar Memcached, la opción de utilizar la clave administrada por el cliente para el cifrado en reposo no se encuentra disponible en las regiones de AWS GovCloud (us-gov-east-1 y us-gov-west-1).

Implementar el cifrado en reposo puede reducir el rendimiento durante las operaciones de backup y sincronización de nodos. Compare sus datos con y sin cifrado en reposo para determinar el impacto en el rendimiento de la implementación.

Uso de claves administradas por el cliente desde AWS KMS

ElastiCache admite las claves de AWS KMS (clave de KMS) simétricas administradas por el cliente para el cifrado en reposo. Las claves de KMS administradas por el cliente son claves de cifrado que crea, posee y administra en la cuenta de AWS. Para obtener más información, consulte Claves de AWS KMS en la Guía para desarrolladores de AWS Key Management Service. Las claves deben crearse en AWS KMS para poder utilizarlas con ElastiCache.

Para obtener más información sobre la creación de claves maestras de AWS KMS, consulte Creación de claves en la Guía para desarrolladores de AWS Key Management Service.

ElastiCache permite la integración con AWS KMS. Para obtener más información, consulte Uso de concesiones en la Guía para desarrolladores de AWS Key Management Service. No se requieren acciones del cliente para habilitar la integración de Amazon ElastiCache con AWS KMS.

La clave de condición `kms:ViaService` limita el uso de una clave de AWS KMS (clave de KMS) en solicitudes de servicios de AWS específicos. Para utilizar `kms:ViaService` con ElastiCache, incluya ambos nombres de ViaService en el valor de clave de condición: `elasticache.AWS_region.amazonaws.com` y `dax.AWS_region.amazonaws.com`. Para obtener más información, consulte kms:ViaService.

Puede utilizar AWS CloudTrail para realizar un seguimiento de las solicitudes que Amazon ElastiCache envía a AWS Key Management Service en su nombre. Todas las llamadas a la API a AWS Key Management Service relacionadas con claves administradas por el cliente tienen los registros de CloudTrail correspondientes. También puede ver las concesiones que crea ElastiCache con la llamada a la llamada a la API de KMS ListGrants.

Una vez que se cifra un grupo de reproducción mediante la clave administrada por el cliente, todas las copias de seguridad para el grupo de reproducción se cifran de la siguiente manera:

- Las copias de seguridad diarias automáticas se cifran mediante la clave administrada por el cliente asociada con el clúster.
- La copia de seguridad final creada cuando se elimina el grupo de reproducción también se cifra mediante la clave administrada por el cliente asociada con el grupo de reproducción.
- Las copias de seguridad creadas de forma manual se cifran de manera predeterminada para utilizar la clave de KMS asociada con el grupo de reproducción. Puede anular esto al elegir otra clave administrada por el cliente.
- Al copiar una copia de seguridad se utiliza de forma predeterminada una clave administrada por el cliente asociada a la copia de seguridad de origen. Puede anular esto al elegir otra clave administrada por el cliente.

> **Note**
>
> - Las claves administradas por el cliente no se pueden utilizar cuando se exportan copias de seguridad al bucket de Amazon S3 seleccionado. Sin embargo, todas las copias de seguridad exportadas a Amazon S3 se cifran mediante el cifrado del lado del servidor. Puede optar por copiar el archivo de copia de seguridad en un objeto de S3 nuevo y cifrarlo mediante una clave de KMS administrada por el cliente, copiar el archivo a otro bucket de S3 que se ha configurado con el cifrado predeterminado mediante una clave de KMS o cambiar una opción de cifrado en el propio archivo.
> - También puede utilizar claves administradas por el cliente a fin de cifrar copias de seguridad creadas de forma manual para grupos de reproducción que no utilicen claves administradas por el cliente para el cifrado. Con esta opción, el archivo de copia de seguridad almacenado en Amazon S3 se cifra mediante una clave de KMS, aunque los datos no se cifren en el grupo de reproducción original.
>
> La restauración desde una copia de seguridad le permite elegir entre las opciones de cifrado disponibles, similares a las opciones de cifrado disponibles cuando se crea un nuevo grupo de reproducción.

- Si elimina la clave o deshabilita la clave y revoca las concesiones para la clave que utilizó para cifrar una caché, esta se vuelve irrecuperable. En otras palabras, no se puede modificar ni recuperar después de un error de hardware. AWS KMS solo elimina las claves maestras después

de un periodo de espera de al menos siete días. Después de eliminar la clave, puede utilizar una clave administrada por el cliente diferente para crear una copia de seguridad con fines de archivo.

- La rotación automática de claves conserva las propiedades de las claves maestras de AWS KMS, por lo que la rotación no tiene efecto sobre la capacidad de acceder a los datos de ElastiCache. Las cachés de Amazon ElastiCache cifradas no admiten la rotación de claves manual, lo que implica la creación de una clave maestra nueva y la actualización de cualquier referencia a la clave antigua. Para obtener más información, consulte Rotación de claves de AWS KMS en la Guía para desarrolladores de AWS Key Management Service.
- El cifrado de una caché de ElastiCache mediante la clave de KMS requiere una concesión por caché. Esa concesión se utiliza durante toda la vida útil de la caché. Además, se utiliza una concesión por copia de seguridad durante la creación de la copia de seguridad. Dicha concesión se retira una vez que se crea la copia de seguridad.
- Para obtener más información sobre los límites y las concesiones de AWS KMS, consulte los Límites en la Guía para desarrolladores de AWS Key Management Service.

Activación del cifrado en reposo

Todas las cachés sin servidor tienen activado el cifrado en reposo.

Al crear un clúster basado en nodos, puede habilitar el cifrado en reposo configurando el parámetro `AtRestEncryptionEnabled` en `true`. No puede habilitar el cifrado en reposo en grupos de reproducción existentes.

Puede habilitar el cifrado en reposo al crear una caché de ElastiCache. Para ello, utilice la Consola de administración de AWS, la AWS CLI o la API de ElastiCache.

Cuando cree una caché, puede elegir una de las siguientes opciones:

- Default (Predeterminado): esta opción utiliza el cifrado administrado por el servicio en reposo.
- Customer managed key (clave administrada por el cliente): esta opción permite proporcionar el ID/ARN de clave desde AWS KMS para el cifrado en reposo.

Para obtener más información sobre la creación de claves maestras de AWS KMS, consulte Crear claves en la Guía para desarrolladores de AWS Key Management Service.

Contenido

- Activación del cifrado en reposo a través de Consola de administración de AWS

- Activación del cifrado en reposo a través de AWS CLI

Habilitación del cifrado en reposo en un clúster de Valkey o Redis OSS basado en nodos existente

El cifrado en reposo solamente se puede habilitar al crear un grupo de replicación de Valkey o Redis OSS. Si tiene un grupo de reproducción existente en el que desea habilitar el cifrado en reposo, haga lo siguiente.

Para habilitar el cifrado en reposo en un grupo de reproducción existente

1. Cree un backup manual del grupo de reproducción existente. Para obtener más información, consulte Copias de seguridad manuales.
2. Cree un nuevo grupo de reproducción a partir de este backup. En el nuevo grupo de reproducción, habilite el cifrado en reposo. Para obtener más información, consulte Restauración desde una copia de seguridad a una nueva caché.
3. Actualice los puntos de enlace de la aplicación para que apunten al nuevo grupo de reproducción.
4. Elimine el grupo de reproducción antiguo. Para obtener más información, consulte Eliminación de un clúster en ElastiCache o Eliminación de un grupo de reproducción.

Activación del cifrado en reposo a través de Consola de administración de AWS

Habilitación del cifrado en reposo en una caché sin servidor (consola)

Todas las cachés sin servidor tienen activado el cifrado en reposo. La clave KMS propiedad de AWS se utiliza de manera predeterminada para cifrar los datos. Para elegir su propia clave AWS KMS, elija las siguientes opciones:

- Amplíe la sección Configuración predeterminada.
- Seleccione Personalizar la configuración predeterminada en la sección Configuración predeterminada.
- Seleccione Personalice su configuración de seguridad en la sección Seguridad.
- Elija CMK administrada por el cliente en Configuración de clave de cifrado.
- Seleccione una clave en el ajuste Clave de AWS KMS.

Activación del cifrado en reposo en un clúster basado en nodos (consola)

Al diseñar su propia caché, las configuraciones de “desarrollo y prueba” y “producción” con el método “Creación sencilla” habilitan el cifrado en reposo mediante la clave Predeterminada. Al elegir la configuración, seleccione estas opciones:

- Seleccione la versión 3.2.6, 4.0.10 o posterior como versión del motor.
- Haga clic en la casilla de verificación situada junto a Habilitar en la opción Cifrado en reposo.
- Elija una Clave predeterminada o una CMK administrada por el cliente.

Para ver el procedimiento paso por paso, consulte:

- Creación de un clúster de Valkey (modo de clúster deshabilitado) (consola)
- Creación de un clúster de Valkey o Redis OSS (modo de clúster habilitado) (consola)

Activación del cifrado en reposo a través de AWS CLI

Para habilitar el cifrado en reposo al crear un clúster de Valkey o Redis OSS con la AWS CLI, utilice el parámetro --at-rest-encryption-enabled cuando cree el grupo de replicación.

Habilitación del cifrado en reposo en un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) (CLI)

La siguiente operación crea el grupo de replicación `my-classic-rg` de Valkey o Redis OSS (modo de clúster deshabilitado) con tres nodos (--num-cache-clusters), un nodo principal y dos réplicas de lectura. El cifrado en reposo está habilitado en este grupo de reproducción (--at-rest-encryption-enabled).

Los siguientes parámetros y sus valores son necesarios para habilitar el cifrado de este grupo de reproducción:

Parámetros clave

- **`--engine`**: debe ser `valkey` o `redis`.
- **`--engine-version`**: si el motor es Redis OSS, debe ser la versión 3.2.6, 4.0.10 o posterior.
- **`--at-rest-encryption-enabled`**: obligatorio para habilitar el cifrado en reposo.

Example 1: clúster de Valkey o Redis OSS (modo de clúster deshabilitado) con réplicas

Para Linux, macOS o Unix:

```
aws elasticache create-replication-group \
    --replication-group-id my-classic-rg \
    --replication-group-description "3 node replication group" \
    --cache-node-type cache.m4.large \
    --engine redis \
    --at-rest-encryption-enabled \
    --num-cache-clusters 3
```

Para Windows:

```
aws elasticache create-replication-group ^
    --replication-group-id my-classic-rg ^
    --replication-group-description "3 node replication group" ^
    --cache-node-type cache.m4.large ^
    --engine redis ^
    --at-rest-encryption-enabled ^
    --num-cache-clusters 3 ^
```

Para obtener más información, consulte los siguientes temas:

- Creación de un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) desde cero (AWS CLI)
- create-replication-group

Habilitación del cifrado en reposo en un clúster para Valkey o Redis OSS (modo de clúster habilitado) (CLI)

La siguiente operación crea el grupo de replicación `my-clustered-rg` de Valkey o Redis OSS (modo de clúster habilitado) con tres grupos de nodos o particiones (--num-node-groups). Cada uno tiene tres nodos: uno principal y dos réplicas de lectura (--replicas-per-node-group). El cifrado en reposo está habilitado en este grupo de reproducción (--at-rest-encryption-enabled).

Los siguientes parámetros y sus valores son necesarios para habilitar el cifrado de este grupo de reproducción:

Parámetros clave

- **`--engine`**: debe ser `valkey` o `redis`.
- **`--engine-version`**: si el motor es Redis OSS, debe ser la versión 4.0.10 o posterior.
- **`--at-rest-encryption-enabled`**: obligatorio para habilitar el cifrado en reposo.
- **`--cache-parameter-group`**: debe ser `default-redis4.0.cluster.on` o un valor derivado de este que lo convierta en un grupo de reproducción con el modo de clúster habilitado.

Example 2: un clúster de Valkey o Redis OSS (modo de clúster habilitado)

Para Linux, macOS o Unix:

```
aws elasticache create-replication-group \
   --replication-group-id my-clustered-rg \
   --replication-group-description "redis clustered cluster" \
   --cache-node-type cache.m3.large \
   --num-node-groups 3 \
   --replicas-per-node-group 2 \
   --engine redis \
   --engine-version 6.2 \
   --at-rest-encryption-enabled \
   --cache-parameter-group default.redis6.x.cluster.on
```

Para Windows:

```
aws elasticache create-replication-group ^
   --replication-group-id my-clustered-rg ^
   --replication-group-description "redis clustered cluster" ^
   --cache-node-type cache.m3.large ^
   --num-node-groups 3 ^
   --replicas-per-node-group 2 ^
   --engine redis ^
   --engine-version 6.2 ^
   --at-rest-encryption-enabled ^
   --cache-parameter-group default.redis6.x.cluster.on
```

Para obtener más información, consulte los siguientes temas:

- Creación de un grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado) desde cero (AWS CLI)

- create-replication-group

Véase también

- Seguridad de ElastiCache y Amazon VPC
- Identity and Access Management para Amazon ElastiCache

Autenticación y autorización

AWS Identity and Access Management (IAM) es un servicio web que lo ayuda a controlar de forma segura el acceso a los recursos de AWS. ElastiCache permite la autenticación de usuarios mediante IAM y el comando AUTH de Valkey y Redis OSS, así como la autorización de operaciones de usuarios mediante el control de acceso basado en roles (RBAC).

Temas

- Control de acceso basado en roles (RBAC)
- Autenticación con el comando AUTH de Valkey o Redis OSS
- Deshabilitación del control de acceso en una caché de ElastiCache Valkey o Redis OSS

Control de acceso basado en roles (RBAC)

Con el comando AUTH de Valkey y Redis OSS, tal como se describe en Autenticación con el comando AUTH de Valkey o Redis OSS, puede utilizar el control de acceso basado en roles (RBAC). El RBAC también es la única forma de controlar el acceso a las cachés sin servidor. Está disponible para Valkey 7.2 en adelante y Redis OSS 6.0 a 7.2.

RBAC le permite:

- Controlar el acceso a la caché a través de grupos de usuarios. Estos grupos de usuarios se han diseñado como una forma de organizar el acceso a las cachés.
- Con authN, puede utilizar contraseñas por usuario en lugar de tokens de autenticación por clúster.
- Con authZ, tiene permisos de usuario detallados.
- Basar su acceso al clúster en ACL.

A diferencia de AUTH de Valkey y Redis OSS, donde todos los clientes autenticados tienen acceso completo a la caché si su token está autenticado, RBAC le permite asignar usuarios a conjuntos en

función de los roles deseados de los usuarios. Estos conjuntos se han diseñado para organizar el acceso a las cachés.

Con el RBAC, puede crear usuarios y asignarles permisos específicos mediante una cadena de acceso, como se describe a continuación. Asigne los usuarios a conjuntos correspondientes a un rol específico (administradores, personal de Recursos Humanos) que luego se implementan en una o más cachés de ElastiCache. Al hacerlo, puede establecer límites de seguridad entre clientes que utilicen las mismas cachés de Valkey o Redis OSS e impedir que los clientes accedan a los datos de los demás.

RBAC se ha diseñado para facilitar la introducción de ACL en Redis OSS 6. Cuando utiliza RBAC con la caché de ElastiCache Valkey o Redis OSS, existen algunas limitaciones que debe tener en cuenta:

- Un grupo de usuarios configurado para el motor VALKEY solo puede contener usuarios que utilicen un mecanismo de autenticación (contraseña o IAM). Esto significa que todos los usuarios con el motor VALKEY y cualquier otro usuario con el motor Redis que tenga su configuración configurada para autenticarse con contraseña o IAM, pueden estar en este grupo de usuarios.
- Cuando se utiliza RBAC con clústeres de Valkey, se pueden utilizar tanto los grupos de usuarios con el motor VALKEY como con el motor REDIS.
- Cuando se utiliza RBAC con clústeres de Redis OSS, solo se pueden utilizar grupos de usuarios con el motor REDIS.
- No puede especificar contraseñas en una cadena de acceso. Establece las contraseñas mediante las llamadas CreateUser o ModifyUser.
- Para los derechos de usuarios, debe activar o desactivar usuarios con `on` y `off` como parte de la cadena de acceso. Si no se especifica ninguno en la cadena de acceso, se le asigna `off` al usuario y no tiene derechos de acceso a la caché.
- No puede utilizar comandos prohibidos o renombrados en una cadena de acceso. Si especifica un comando prohibido o renombrado, se generará una excepción. Si desea utilizar listas de control de acceso (ACL) para un comando renombrado, especifique el nombre original del comando, es decir, el nombre del comando antes de cambiarle el nombre.
- No puede utilizar el comando `reset` como parte de una cadena de acceso. Las contraseñas se especifican con parámetros de API y ElastiCache para Valkey y Redis OSS administra las contraseñas. Por lo tanto, no puede utilizar `reset` porque eliminará todas las contraseñas de un usuario.

- Redis OSS 6 introduce el comando ACL LIST. Este comando devuelve una lista de usuarios junto con las reglas de ACL aplicadas a cada usuario. ElastiCache admite el comando `ACL LIST`, pero no admite hash de contraseña como lo hace Redis OSS. Con ElastiCache, puede utilizar la operación DescribeUsers para obtener información similar, incluidas las reglas que se encuentran en la cadena de acceso. Sin embargo, DescribeUsers no recupera una contraseña de usuario.
- Otros comandos de solo lectura que admite ElastiCache para Valkey y Redis OSS son ACL WHOAMI, ACL USERS y ACL CAT, entre otros. ElastiCache para Valkey y Redis OSS no admite otros comandos de ACL basados en escritura.
- Se aplican los siguientes límites:

Recurso	Máximo permitido
Usuarios por grupo de usuarios	100
Número de usuarios	1000
Número de grupos de usuarios	100

RBAC con Valkey

Cuando se utiliza el control de acceso basado en roles con Valkey, los usuarios y los grupos de usuarios se crean con el tipo de motor VALKEY. Se recomienda esta opción, ya que, de forma predeterminada, Valkey con RBAC proporciona mayor seguridad en comparación con Redis OSS. Tanto los clústeres de Valkey aprovisionados como sin servidor admiten las asociaciones de usuarios y grupos de usuarios VALKEY.

Las principales características del control de acceso a Valkey son:

- Los usuarios de Valkey están restringidos únicamente a las asociaciones de grupos de usuarios de Valkey.
- Los grupos de usuarios de Valkey pueden incluir usuarios de Valkey y usuarios de Redis OSS protegidos con contraseña o habilitados para la autenticación de IAM.
- Los usuarios de Valkey deben utilizar la protección por contraseña o la autenticación de IAM.
- Los grupos de usuarios VALKEY solo se pueden asociar con los clústeres VALKEY.

- No hay ningún requisito de usuario predeterminado. Cuando el grupo de usuarios de Valkey está asociado con clústeres, el requisito de usuario predeterminado se desactiva automáticamente. Los clientes verán que el usuario predeterminado está desactivado al usar el comando ACL LIST.

A continuación, encontrará más información sobre el uso de RBAC con ElastiCache para Valkey y Redis OSS.

Temas

- Especificación de permisos mediante una cadena de acceso
- Aplicación de RBAC a una caché para ElastiCache para Valkey o Redis OSS
- Migración de AUTH a RBAC
- Migración de RBAC a AUTH
- Rotación automática de contraseñas para usuarios
- Autenticación con IAM

Especificación de permisos mediante una cadena de acceso

Para especificar permisos para una caché de ElastiCache para Valkey o Redis OSS, debe crear una cadena de acceso y asignarla a un usuario mediante la AWS CLI o la Consola de administración de AWS.

Las cadenas de acceso se definen como una lista de reglas delimitadas por espacios que se aplican al usuario. Definen qué comandos puede ejecutar un usuario y qué claves puede operar. Para ejecutar un comando, un usuario debe tener acceso al comando que se ejecuta y a todas las claves a las que accede el comando. Las reglas se aplican de izquierda a derecha de forma acumulativa y se puede utilizar una cadena más simple en lugar de la proporcionada si hay redundancias en la cadena proporcionada.

Para obtener más información sobre la sintaxis de las reglas de ACL, consulte ACL.

En el siguiente ejemplo, la cadena de acceso representa un usuario activo con acceso a todas las claves y comandos disponibles.

```
on ~* +@all
```

La sintaxis de la cadena de acceso se desglosa de la siguiente manera:

- on: el usuario es un usuario activo.

- `~*`: se brinda acceso a todas las claves disponibles.
- `+@all`: se brinda acceso a todos los comandos disponibles.

La configuración anterior es la menos restrictiva. Puede modificar esta configuración para hacerla más segura.

En el siguiente ejemplo, la cadena de acceso representa a un usuario con acceso restringido al acceso de lectura en claves que comienzan por el espacio de claves “app::”

```
on ~app::* -@all +@read
```

Puede refinar aún más estos permisos al enumerar comandos a los que el usuario tiene acceso:

`+`*`command1`* – el acceso del usuario a los comandos se encuentra limitado a *`command1`*.

`+@category`: el acceso del usuario a los comandos se encuentra limitado a la categoría de comandos.

Para obtener información sobre cómo asignar una cadena de acceso a un usuario, consulte Creación de usuarios y grupos de usuarios con la consola y la CLI.

Si va a migrar una carga de trabajo existente a ElastiCache, puede recuperar la cadena de acceso mediante una llamada a `ACL LIST`, que excluya el usuario y cualquier hash de contraseña.

Para la versión 6.2 y posteriores de Redis OSS, también se admite la siguiente sintaxis de cadena de acceso:

- `&*`: se brinda acceso a todos los canales disponibles.

Para la versión 7.0 y posteriores de Redis OSS, también se admite la siguiente sintaxis de cadena de acceso:

- `|`: se puede usar para bloquear subcomandos (por ejemplo, “-config|set”).
- `%R~<pattern>`: agregue el patrón de claves de lectura especificado. Esto se comporta de forma similar al patrón de claves normal, pero solo permite leer las claves que coinciden con el patrón dado. Consulte key permissions (permisos de clave) para obtener más información.
- `%W~<pattern>`: agregue el patrón de claves de escritura especificado. Esto se comporta de forma similar al patrón de claves normal, pero solo concede permiso de escritura a las claves que coinciden con el patrón dado. Consulte ACL key permissions para obtener más información.

- `%RW~<pattern>`: alias para `~<pattern>`.
- `(<rule list>)`: cree un nuevo selector con el que comparar las reglas. Los selectores se evalúan después de los permisos del usuario y según el orden en que se definen. Si un comando coincide con los permisos del usuario o con algún selector, está permitido. Consulte la página sobre selectores de ACL para obtener más información.
- `clearselectors`: elimine todos los selectores asociados al usuario.

Aplicación de RBAC a una caché para ElastiCache para Valkey o Redis OSS

Para utilizar RBAC de ElastiCache para Valkey o Redis OSS, siga estos pasos:

1. Cree uno o más usuarios.
2. Cree un grupo de usuarios y agregue usuarios al grupo.
3. Asigne el grupo de usuarios a una caché que tenga el cifrado en tránsito habilitado.

Estos pasos se describen con detalle a continuación.

Temas

- Creación de usuarios y grupos de usuarios con la consola y la CLI
- Administración de grupos de usuarios con la consola y la CLI
- Asignación de grupos de usuarios a cachés sin servidor
- Asignación de grupos de usuarios a grupos de reproducción

Creación de usuarios y grupos de usuarios con la consola y la CLI

La información de usuario para los usuarios de RBAC es un ID de usuario, nombre de usuario y, opcionalmente, una contraseña y una cadena de acceso. La cadena de acceso proporciona el nivel de permisos en las claves y comandos. El ID de usuario es exclusivo del usuario y el nombre de usuario es lo que se pasa al motor.

Asegúrese de que los permisos de usuario que proporcione tengan sentido con el propósito previsto del grupo de usuarios. Por ejemplo, si crea un grupo de usuarios denominado `Administrators`, cualquier usuario que agregue a ese grupo debe tener su cadena de acceso establecida en el acceso completo a las claves y comandos. Para los usuarios de un grupo de usuarios `e-commerce`, puede establecer las cadenas de acceso en acceso de solo lectura.

ElastiCache configura de forma automática un usuario predeterminado con el ID de usuario y el nombre de usuario `"default"` y lo agrega a todos los grupos de usuarios. No puede modificar ni eliminar este usuario. Este usuario se ha diseñado para ser compatible con el comportamiento predeterminado de las versiones anteriores de Redis OSS y tiene una cadena de acceso que permite llamar a todos los comandos y acceder a todas las claves.

Para agregar un control de acceso adecuado a una caché, reemplace este usuario predeterminado por uno nuevo que no se encuentre habilitado o utilice una contraseña segura. Para cambiar el usuario predeterminado, cree un usuario nuevo con el nombre de usuario establecido en `default`. A continuación, puede intercambiarlo con el usuario predeterminado original.

En los siguientes procedimientos se muestra cómo intercambiar el usuario `default` original con otro usuario `default` que tiene una cadena de acceso modificada.

Para modificar el usuario predeterminado de la consola

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. Seleccione Administración de grupos de usuarios en el panel de navegación.
3. En ID de grupo de usuarios, elija el ID que desea modificar. Asegúrese de elegir el enlace y no la casilla de verificación.
4. Elija Modificar.
5. En la ventana Modificar, seleccione Administrar. En Seleccione el usuario que desee, elija el usuario con el Nombre de usuario predeterminado.
6. Seleccione Elegir.
7. Elija Modificar. Al hacerlo, se terminan todas las conexiones existentes que el usuario predeterminado tenga a una caché.

Para modificar el usuario predeterminado con la AWS CLI

1. Cree un usuario nuevo con el nombre de usuario `default` mediante los siguientes comandos.

   Para Linux, macOS o Unix:

```
aws elasticache create-user \
 --user-id "new-default-user" \
 --user-name "default" \
 --engine "VALKEY" \
```

```
 --passwords "a-str0ng-pa))word" \
 --access-string "off +get ~keys*"
```

Para Windows:

```
aws elasticache create-user ^
 --user-id "new-default-user" ^
 --user-name "default" ^
 --engine "VALKEY" ^
 --passwords "a-str0ng-pa))word" ^
 --access-string "off +get ~keys*"
```

2. Cree un grupo de usuarios y agregue el usuario que creó anteriormente.

   Para Linux, macOS o Unix:

```
aws elasticache create-user-group \
  --user-group-id "new-group-2" \
  --engine "VALKEY" \
  --user-ids "new-default-user"
```

   Para Windows:

```
aws elasticache create-user-group ^
  --user-group-id "new-group-2" ^
  --engine "VALKEY" ^
  --user-ids "new-default-user"
```

Al crear un usuario, puede configurar hasta dos contraseñas. Al modificar una contraseña, se mantienen todas las conexiones existentes a las cachés.

En concreto, tenga en cuenta estas restricciones de contraseña del usuario al utilizar RBAC con ElastiCache para Valkey y Redis OSS:

- Las contraseñas deben tener entre 16 y 128 caracteres imprimibles.
- No se admiten los siguientes caracteres no alfanuméricos: `, "" / @`.

Administración de usuarios con la consola y la CLI

Utilice el siguiente procedimiento para administrar usuarios en la consola.

Para administrar usuarios con la consola

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de control de Amazon ElastiCache, elija Administración de usuarios. Están disponibles las siguientes opciones:

   - Crear usuario: al crear un usuario, introduce un ID de usuario, un nombre de usuario, un modo de autenticación y una cadena de acceso. La cadena de acceso establece el nivel de permisos para qué claves y comandos se permite al usuario.

     Al crear un usuario, puede configurar hasta dos contraseñas. Al modificar una contraseña, se mantienen todas las conexiones existentes a las cachés.
   - Modificar usuario: permite actualizar la configuración de autenticación de un usuario o cambiar su cadena de acceso.
   - Eliminar usuario: la cuenta se eliminará de cualquier grupo de usuarios al que pertenezca.

Utilice los siguientes procedimientos para administrar usuarios con la AWS CLI.

Para modificar un usuario mediante la CLI

- Utilice el comando `modify-user` para actualizar la contraseña o contraseñas de un usuario o cambiar los permisos de acceso de un usuario.

  Cuando se modifica un usuario, se actualizan los grupos de usuarios asociados al usuario, junto con las cachés asociadas al grupo de usuarios. Se mantienen todas las conexiones existentes. A continuación se muestran algunos ejemplos.

  Para Linux, macOS o Unix:

  ```
  aws elasticache modify-user \
    --user-id user-id-1 \
    --access-string "~objects:* ~items:* ~public:*" \
    --authentication-mode Type=iam
  ```

Para Windows:

```
aws elasticache modify-user ^
  --user-id user-id-1 ^
  --access-string "~objects:* ~items:* ~public:*" ^
  --authentication-mode Type=iam
```

> **Note**
>
> No recomendamos utilizar la opción `nopass`. Si lo hace, recomendamos que configure los permisos del usuario en solo lectura con acceso a un conjunto limitado de claves.

Para eliminar un usuario mediante la CLI

- Ejecute el comando `delete-user` para eliminar un usuario. La cuenta se borra y elimina de todos los grupos de usuarios a los que pertenezca. A continuación se muestra un ejemplo.

  Para Linux, macOS o Unix:

  ```
  aws elasticache delete-user \
    --user-id user-id-2
  ```

  Para Windows:

  ```
  aws elasticache delete-user ^
    --user-id user-id-2
  ```

Para ver una lista de usuarios, llame a la operación describe-users.

```
aws elasticache describe-users
```

Administración de grupos de usuarios con la consola y la CLI

Puede crear grupos de usuarios para organizar y controlar el acceso de los usuarios a una o más cachés, como se muestra a continuación.

Utilice el siguiente procedimiento para administrar grupos de usuarios en la consola.

Para administrar grupos de usuarios mediante la consola

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de control de Amazon ElastiCache, elija Administración de grupos de usuarios.

   A continuación, se muestran las operaciones disponibles para crear grupos de usuarios nuevos:

   - Crear: al crear un grupo de usuarios, se agregan usuarios y, a continuación, se asignan los grupos de usuarios a las cachés. Por ejemplo, puede crear un grupo de usuarios `Admin` para los usuarios que tienen roles administrativos en una caché.

     > **Important**
     >
     > Si no utiliza ningún grupo de usuarios de Valkey o Redis OSS, debe incluir un usuario predeterminado al crear un grupo de usuarios.

   - Agregar usuarios: agregue usuarios al grupo de usuarios.
   - Eliminar usuarios: elimine usuarios del grupo de usuarios. Cuando se quitan usuarios de un grupo de usuarios, se terminan todas las conexiones existentes que tengan a una caché.
   - Eliminar: utilice esta opción para eliminar un grupo de usuarios. Tenga en cuenta que se eliminará el grupo de usuarios en sí, no los usuarios que pertenecen al grupo.

   Para los grupos de usuarios existentes, puede realizar lo siguiente:

   - Agregar usuarios: agregue usuarios existentes al grupo de usuarios.
   - Eliminar usuarios: elimine los usuarios existentes del grupo de usuarios.

     > **Note**
     >
     > Los usuarios se eliminan del grupo de usuarios, pero no se eliminan del sistema.

Utilice el siguiente procedimiento para administrar grupos de usuarios en la CLI.

Para crear un grupo de usuarios nuevo y agregar un usuario mediante la CLI

- Utilice el comando `create-user-group`, tal y como se muestra a continuación:

  Para Linux, macOS o Unix:

```
aws elasticache create-user-group \
  --user-group-id "new-group-1" \
  --engine "VALKEY" \
  --user-ids user-id-1, user-id-2
```

  Para Windows:

```
aws elasticache create-user-group ^
  --user-group-id "new-group-1" ^
  --engine "VALKEY" ^
  --user-ids user-id-1, user-id-2
```

Para modificar un grupo de usuarios al agregar usuarios nuevos o eliminar miembros actuales mediante la CLI

- Utilice el comando `modify-user-group`, tal y como se muestra a continuación:

  Para Linux, macOS o Unix:

```
aws elasticache modify-user-group --user-group-id new-group-1 \
--user-ids-to-add user-id-3 \
--user-ids-to-remove user-id-2
```

  Para Windows:

```
aws elasticache modify-user-group --user-group-id new-group-1 ^
--user-ids-to-add user-id-3 ^
--user-ids-to-removere user-id-2
```

> **Note**
>
> Cualquier conexión abierta que pertenezca a un usuario eliminado de un grupo de usuarios finaliza con este comando.

Para eliminar un grupo de usuarios mediante la CLI

- Utilice el comando `delete-user-group`, tal y como se muestra a continuación: Se elimina el grupo de usuarios en sí, no los usuarios que pertenecen al grupo.

  Para Linux, macOS o Unix:

  ```
  aws elasticache delete-user-group /
     --user-group-id
  ```

  Para Windows:

  ```
  aws elasticache delete-user-group ^
     --user-group-id
  ```

Para ver una lista de grupos de usuarios, puede llamar a la operación describe-user-groups.

```
aws elasticache describe-user-groups \
  --user-group-id test-group
```

Asignación de grupos de usuarios a cachés sin servidor

Después de crear un grupo de usuarios y agregar usuarios, el paso final para implementar el RBAC es asignar el grupo de usuarios a una caché sin servidor.

Asignación de grupos de usuarios a cachés sin servidor mediante la consola

Para agregar un grupo de usuarios a una caché sin servidor mediante la Consola de administración de AWS, haga lo siguiente:

- Para el modo de clúster deshabilitado, consulte Creación de un clúster de Valkey (modo de clúster deshabilitado) (consola)

- Para el modo de clúster habilitado, consulte Creación de un clúster de Valkey o Redis OSS (modo de clúster habilitado) (consola)

Asignación de grupos de usuarios a cachés sin servidor mediante la AWS CLI

La siguiente operación de la AWS CLI crea una caché sin servidor mediante el parámetro user-group-id con el valor *my-user-group-id*. Reemplace el grupo de subredes `sng-test` por otro existente.

Parámetros clave

- **`--engine`**: debe ser `VALKEY` o `REDIS`.
- **`--user-group-id`**: este valor proporciona el ID del grupo de usuarios, compuesto por usuarios con permisos de acceso especificados para la caché.

Para Linux, macOS o Unix:

```
aws elasticache create-serverless-cache \
    --serverless-cache-name "new-serverless-cache" \
    --description "new-serverless-cache" \
    --engine "VALKEY" \
    --user-group-id "new-group-1"
```

Para Windows:

```
aws elasticache create-serverless-cache ^
    --serverless-cache-name "new-serverless-cache" ^
    --description "new-serverless-cache" ^
    --engine "VALKEY" ^
    --user-group-id "new-group-1"
```

La siguiente operación de la AWS CLI modifica una caché sin servidor mediante el parámetro user-group-id con el valor *my-user-group-id*.

Para Linux, macOS o Unix:

```
aws elasticache modify-serverless-cache \
    --serverless-cache-name serverless-cache-1 \
    --user-group-id "new-group-2"
```

Para Windows:

```
aws elasticache modify-serverless-cache ^
    --serverless-cache-name serverless-cache-1 ^
    --user-group-id "new-group-2"
```

Tenga en cuenta que cualquier modificación que se haga en una caché se actualizará de forma asíncrona. Puede monitorear el progreso al visualizar los eventos. Para obtener más información, consulte Visualización de eventos de ElastiCache.

Asignación de grupos de usuarios a grupos de reproducción

Después de crear un grupo de usuarios y agregar usuarios, el paso final para implementar el RBAC es asignar el grupo de usuarios a un grupo de reproducción.

Asignación de grupos de usuarios a grupos de reproducción mediante la consola

Para agregar un grupo de usuarios a una reproducción mediante la Consola de administración de AWS, realice lo siguiente:

- Para el modo de clúster deshabilitado, consulte Creación de un clúster de Valkey (modo de clúster deshabilitado) (consola)
- Para el modo de clúster habilitado, consulte Creación de un clúster de Valkey o Redis OSS (modo de clúster habilitado) (consola)

Asignación de grupos de usuarios a grupos de reproducción mediante la AWS CLI

La siguiente operación de la AWS CLI crea un grupo de reproducción con el cifrado en tránsito (TLS) habilitado y el parámetro user-group-ids con el valor *`my-user-group-id`*. Reemplace el grupo de subredes `sng-test` por otro existente.

Parámetros clave

- **`--engine`**: debe ser `valkey` o `redis`.
- **`--engine-version`**: debe ser 6.0 o posterior.
- **`--transit-encryption-enabled`**: obligatorio para la autenticación y a fin de asociar un grupo de usuarios.
- **`--user-group-ids`**: este valor proporciona el ID del grupo de usuarios, compuesto por usuarios con permisos de acceso especificados para la caché.

- **`--cache-subnet-group`**: obligatorio para asociar un grupo de usuarios.

Para Linux, macOS o Unix:

```
aws elasticache create-replication-group \
    --replication-group-id "new-replication-group" \
    --replication-group-description "new-replication-group" \
    --engine "VALKEY" \
    --cache-node-type cache.m5.large \
    --transit-encryption-enabled \
    --user-group-ids "new-group-1" \
    --cache-subnet-group "cache-subnet-group"
```

Para Windows:

```
aws elasticache create-replication-group ^
    --replication-group-id "new-replication-group" ^
    --replication-group-description "new-replication-group" ^
    --engine "VALKEY" ^
    --cache-node-type cache.m5.large ^
    --transit-encryption-enabled ^
    --user-group-ids "new-group-1" ^
    --cache-subnet-group "cache-subnet-group"
```

La siguiente operación de la AWS CLI modifica un grupo de reproducción con el cifrado en tránsito (TLS) habilitado y el parámetro user-group-ids con el valor *`my-user-group-id`*.

Para Linux, macOS o Unix:

```
aws elasticache modify-replication-group \
    --replication-group-id replication-group-1 \
    --user-group-ids-to-remove "new-group-1" \
    --user-group-ids-to-add "new-group-2"
```

Para Windows:

```
aws elasticache modify-replication-group ^
    --replication-group-id replication-group-1 ^
    --user-group-ids-to-remove "new-group-1" ^
    --user-group-ids-to-add "new-group-2"
```

Tenga en cuenta los `PendingChanges` en la respuesta. Las modificaciones que se hagan en una caché se actualizarán de forma asíncrona. Puede monitorear el progreso al visualizar los eventos. Para obtener más información, consulte Visualización de eventos de ElastiCache.

Migración de AUTH a RBAC

Si se encuentra utilizando AUTH como se describe en Autenticación con el comando AUTH de Valkey o Redis OSS y desea migrar a RBAC, utilice los siguientes procedimientos.

Utilice el siguiente procedimiento para migrar de AUTH a RBAC con la consola.

Migración de AUTH de Valkey o Redis OSS a RBAC mediante la consola

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En la lista de la esquina superior derecha, elija la región de AWS en la que se encuentra la caché que desea modificar.
3. En el panel de navegación, elija el motor que se ejecuta en la caché que desea modificar.

   Aparecerá una lista de las cachés del motor elegido.
4. En la lista de cachés, elija el nombre de la caché que desea modificar.
5. Para Actions (Acciones), elija Modify (Modificar).

   Aparecerá la ventana Modificar.
6. En Opción de control de acceso, elija Lista de control de acceso de grupos de usuarios.
7. En Lista de control de acceso de grupos de usuarios, elija un grupo de usuarios.
8. Seleccione Vista previa de los cambios y seleccione Modificar en la siguiente pantalla.

Utilice el siguiente procedimiento para migrar de AUTH de Valkey o Redis OSS a RBAC con la CLI.

Migración de AUTH a RBAC mediante la CLI

- Utilice el comando `modify-replication-group`, tal y como se muestra a continuación:

  Para Linux, macOS o Unix:

```
aws elasticache modify-replication-group --replication-group-id test \
    --auth-token-update-strategy DELETE \
```

```
    --user-group-ids-to-add user-group-1
```

Para Windows:

```
aws elasticache modify-replication-group --replication-group-id test ^
    --auth-token-update-strategy DELETE ^
    --user-group-ids-to-add user-group-1
```

Migración de RBAC a AUTH

Si utiliza RBAC y desea migrar a AUTH de Redis OSS, consulte Migración de RBAC a AUTH.

> **Note**
>
> Si necesita deshabilitar el control de acceso en una caché de ElastiCache, deberá hacerlo a través de la AWS CLI. Para obtener más información, consulte the section called "Deshabilitación del control de acceso en una caché de ElastiCache Valkey o Redis OSS".

Rotación automática de contraseñas para usuarios

Con AWS Secrets Manager, puede reemplazar las credenciales codificadas en el código (incluidas contraseñas), con una llamada a la API de Secrets Manager para recuperar el secreto mediante programación. Esto ayuda a garantizar la integridad del secreto si alguien examina el código, dado que el secreto sencillamente no está allí. Asimismo, puede configurar Secrets Manager para rotar el secreto automáticamente de acuerdo con la programación que especifique. Esto le permite reemplazar secretos a largo plazo con secretos a corto plazo, lo que contribuye a reducir significativamente el riesgo de peligro.

Con Secrets Manager, puede rotar automáticamente las contraseñas de ElastiCache para Redis OSS (es decir, secretos) mediante una función de AWS Lambda que proporciona Secrets Manager.

Para obtener más información acerca del AWS Secrets Manager, consulte ¿Qué es el AWS Secrets Manager?

Cómo usa ElastiCache los secretos

Valkey 7.2 y versiones posteriores tiene un conjunto de características equivalente al de Redis OSS 7.0. En Redis OSS 6, ElastiCache introdujo Control de acceso basado en roles (RBAC)

para proteger el clúster de Valkey o Redis OSS. Esta característica permite limitar determinadas conexiones en términos de los comandos que se pueden ejecutar y las claves a las que se puede acceder. Con RBAC, mientras el cliente crea un usuario con contraseñas, los valores de la contraseña se deben ingresar manualmente en texto no cifrado y estar visibles para el operador.

Con Secrets Manager, las aplicaciones obtienen la contraseña de Secrets Manager en lugar de ingresarla manualmente y almacenarla en la configuración de la aplicación. Para obtener información sobre cómo hacerlo, consulte Cómo los usuarios de ElastiCache se asocian con el secreto.

El uso de secretos conlleva un coste. Para obtener información sobre precios, consulte Precios de AWS Secrets Manager.

Cómo los usuarios de ElastiCache se asocian con el secreto

Secrets Manager guardará una referencia del usuario asociado en el campo `SecretString` del secreto. No habrá ninguna referencia al secreto desde el lado de ElastiCache.

```
{
    "password": "strongpassword",
    "username": "user1",
    "user_arn": "arn:aws:elasticache:us-east-1:xxxxxxxxxx918:user:user1" //this is the
 bond between the secret and the user
}
```

Función de rotación de Lambda

Para habilitar la rotación automática de contraseñas de Secrets Manager, se creará una función de Lambda que interactuará con la API modificar-usuario para actualizar las contraseñas del usuario.

Para obtener información sobre cómo funciona esto, consulte Cómo funciona la rotación.

> **Note**
>
> En el caso de algunos servicios de AWS, para evitar un escenario de suplente confuso, AWS recomienda que se utilicen las claves de condición globales `aws:SourceArn` y `aws:SourceAccount`. No obstante, si se incluye la condición `aws:SourceArn` en la política de la función de rotación, la función de rotación solo se puede utilizar para rotar el secreto especificado por ese ARN. Se recomienda incluir solo la clave de contexto `aws:SourceAccount`, para poder utilizar la función de rotación para varios secretos.

Para cualquier problema que pueda surgir, consulte Solucionar problemas con la rotación de AWS Secrets Manager.

Cómo crear un usuario de ElastiCache y asociarlo con Secrets Manager

Los siguientes pasos muestran cómo crear un usuario y asociarlo con Secrets Manager:

1. Crear un usuario inactivo

   Para Linux, macOS o Unix:

```
aws elasticache create-user \
 --user-id user1 \
 --user-name user1 \
 --engine "REDIS" \
 --no-password \ // no authentication is required
 --access-string "*off* +get ~keys*" // this disables the user
```

   Para Windows:

```
aws elasticache create-user ^
 --user-id user1 ^
 --user-name user1 ^
 --engine "REDIS" ^
 --no-password ^ // no authentication is required
 --access-string "*off* +get ~keys*" // this disables the user
```

   Verá una respuesta similar a la siguiente:

```
{
    "UserId": "user1",
    "UserName": "user1",
    "Status": "active",
    "Engine": "redis",
    "AccessString": "off ~keys* -@all +get",
    "UserGroupIds": [],
    "Authentication": {
        "Type": "no_password"
    },
    "ARN": "arn:aws:elasticache:us-east-1:xxxxxxxxxx918:user:user1"
}
```

2. Crear un secreto

   Para Linux, macOS o Unix:

```
aws secretsmanager create-secret \
--name production/ec/user1 \
--secret-string \
'{
   "user_arn": "arn:aws:elasticache:us-east-1:123456xxxx:user:user1",
    "username":"user1"
 }'
```

   Para Windows:

```
aws secretsmanager create-secret ^
--name production/ec/user1 ^
--secret-string ^
'{
   "user_arn": "arn:aws:elasticache:us-east-1:123456xxxx:user:user1",
    "username":"user1"
 }'
```

   Verá una respuesta similar a la siguiente:

```
{
 "ARN": "arn:aws:secretsmanager:us-east-1:123456xxxx:secret:production/ec/user1-
eaFois",
 "Name": "production/ec/user1",
 "VersionId": "aae5b963-1e6b-4250-91c6-ebd6c47d0d95"
}
```

3. Configurar una función de Lambda para rotar la contraseña

   a. Inicie sesión en la Consola de administración de AWS y abra la consola de Lambda en https://console.aws.amazon.com/lambda/

   b. Elija Functions (Funciones) en el panel de navegación y, a continuación, elija la función que ha creado. Elija el nombre de la función, no la casilla de verificación situada a la izquierda.

   c. Elija la pestaña Configuración.

   d. En General configuration (Configuración general), elija Edit (Editar) y, a continuación, establezca Timeout (Tiempo de espera) en al menos 12 minutos.

e. Seleccione Save.

f. Elija Environment variables (Variables de entorno) y, a continuación, establezca lo siguiente:

   i. SECRETS_MANAGER_ENDPOINT: https://secretsmanager.**REGION**.amazonaws.com

   ii. SECRET_ARN: el nombre de recurso de Amazon (ARN) del secreto que creó en el paso 2.

   iii. USER_NAME: nombre de usuario del usuario de ElastiCache,

   iv. Seleccione Save.

g. Elegir Permissions (Permisos)

h. En Execution role (Rol de ejecución), elija el nombre del rol de la función de Lambda para verla en la consola de IAM.

i. La función de Lambda necesitará el siguiente permiso para modificar los usuarios y establecer la contraseña:

   ElastiCache

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:DescribeUsers",
                "elasticache:ModifyUser"
            ],
            "Resource": "arn:aws:elasticache:us-
east-1:123456789012:user:user1"
        }
    ]
}
```

   Secrets Manager

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret",
                "secretsmanager:PutSecretValue",
                "secretsmanager:UpdateSecretVersionStage"
            ],
            "Resource": "arn:aws:secretsmanager:us-
east-1:123456789012:secret:XXXX"
        },
        {
            "Effect": "Allow",
            "Action": "secretsmanager:GetRandomPassword",
            "Resource": "*"
        }
    ]
}
```

4. Configurar la rotación de secretos de Secrets Manager

    a. Mediante la Consola de administración de AWS, consulte Configurar la rotación automática de los secretos de AWS Secrets Manager con la consola

    Para obtener más información sobre la configuración de un programa de rotación, consulte Programación de expresiones en la rotación de Secrets Manager.

    b. Mediante la AWS CLI, consulte Configurar la rotación automática para AWS Secrets Manager mediante AWS Command Line Interface

Autenticación con IAM

Temas

- Descripción general
- Limitaciones

- Configuración
- Conexión

Descripción general

Con la autenticación de IAM, puede autenticar una conexión a ElastiCache para Valkey o Redis OSS mediante identidades de AWS IAM cuando la caché esté configurada para usar la versión 7 o una posterior de Valkey o Redis OSS. Esto le permite reforzar el modelo de seguridad y simplificar muchas tareas de seguridad administrativa. También puede usar la autenticación de IAM a fin de configurar un control de acceso detallado para cada caché y usuario de ElastiCache, con base en los principios de permisos de privilegio mínimo. La autenticación de IAM para ElastiCache se realiza proporcionando un token de autenticación de IAM de corta duración, en lugar de una contraseña de usuario de ElastiCache de larga duración en el comando `AUTH` o `HELLO` de Valkey o Redis OSS. Para obtener más información sobre el token de autenticación de IAM, consulte el proceso de firma de la versión 4 de Signature en la guía de referencia general de AWS y el ejemplo de código que se muestra a continuación.

Puede utilizar las identidades de IAM y sus políticas asociadas para restringir aún más el acceso a Valkey o Redis OSS. También puede conceder acceso a los usuarios desde sus proveedores de identidades federados directamente a las cachés de Valkey o Redis OSS.

Para usar AWS IAM con ElastiCache, primero debe crear un usuario de ElastiCache con el modo de autenticación establecido en IAM. A continuación, puede crear o reutilizar una identidad de IAM. La identidad de IAM necesita una política asociada para conceder la acción `elasticache:Connect` a la caché y el usuario de ElastiCache. Una vez configurado, puede crear un token de autenticación de IAM con las credenciales de AWS del usuario o rol de IAM. Por último, debe proporcionar el token de autenticación de IAM de corta duración como contraseña en el cliente de Valkey o Redis OSS cuando se conecte a la caché. Un cliente de Valkey o Redis OSS compatible con el proveedor de credenciales puede generar automáticamente las credenciales temporales para cada nueva conexión. ElastiCache realizará la autenticación de IAM para las solicitudes de conexión de los usuarios de ElastiCache habilitados para IAM y validará las solicitudes de conexión con IAM.

Limitaciones

Si utiliza la autenticación de IAM, se aplicarán las siguientes limitaciones:

- La autenticación de IAM está disponible cuando se utiliza ElastiCache para Valkey 7.2 y versiones posteriores o Redis OSS 7.0 y versiones posteriores.

- En el caso de los usuarios de ElastiCache habilitados para IAM, las propiedades del ID de nombre de usuario y usuario deben ser idénticas.
- El token de autenticación de IAM es válido durante 15 minutos. Para conexiones de larga duración, recomendamos utilizar un cliente de Valkey o Redis OSS que admita la interfaz de un proveedor de credenciales.
- La conexión autenticada de IAM a ElastiCache para Valkey o Redis OSS se desconectará automáticamente después de 12 horas. La conexión se puede prolongar durante 12 horas enviando un comando `AUTH` o `HELLO` con un nuevo token de autenticación de IAM.
- Los comandos `MULTI EXEC` no admiten la autenticación de IAM.
- Actualmente, la autenticación de IAM admite las siguientes claves de contexto de condición global:
  - Cuando se utiliza la autenticación de IAM con cachés sin servidor, se admiten `aws:VpcSourceIp`, `aws:SourceVpc`, `aws:SourceVpce`, `aws:CurrentTime`, `aws:EpochTime` y `aws:ResourceTag/%s` (desde usuarios y cachés sin servidor asociados).
  - Cuando se utiliza la autenticación de IAM con grupos de réplica, se admiten `aws:SourceIp` y `aws:ResourceTag/%s` (desde los usuarios y grupos de réplica asociados).

  Para obtener más información sobre las claves de contexto de condición globales, consulte Claves de contexto de condición globales de AWS en la Guía del usuario de IAM.

Configuración

Para configurar la autenticación de IAM:

1. Creación de una caché

```
aws elasticache create-serverless-cache \
  --serverless-cache-name cache-01  \
  --description "ElastiCache IAM auth application" \
  --engine redis
```

2. Cree un documento de política de confianza de IAM, como se muestra a continuación, para el rol que permita a la cuenta asumir el nuevo rol. Guarde la política en un archivo denominado trust-policy.json.

   JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "sts:AssumeRole",
        "Resource": "arn:aws:iam::*:role/*"
    }
}
```

3. Cree un documento de política de IAM, como se muestra a continuación. Guarde la política en un archivo denominado policy.json.

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect" : "Allow",
      "Action" : [
        "elasticache:Connect"
      ],
      "Resource" : [
        "arn:aws:elasticache:us-
east-1:123456789012:serverlesscache:cache-01",
        "arn:aws:elasticache:us-east-1:123456789012:user:iam-user-01"
      ]
    }
  ]
}
```

4. Crear un rol de IAM.

```
aws iam create-role \
--role-name "elasticache-iam-auth-app" \
--assume-role-policy-document file://trust-policy.json
```

5. Creación de la política de IAM.

```
aws iam create-policy \
  --policy-name "elasticache-allow-all" \
  --policy-document file://policy.json
```

6. Adjunte la política de IAM al rol.

```
aws iam attach-role-policy \
 --role-name "elasticache-iam-auth-app" \
 --policy-arn "arn:aws:iam::123456789012:policy/elasticache-allow-all"
```

7. Cree un nuevo usuario habilitado para IAM.

```
aws elasticache create-user \
  --user-name iam-user-01 \
  --user-id iam-user-01 \
  --authentication-mode Type=iam \
  --engine redis \
  --access-string "on ~* +@all"
```

8. Cree un grupo de usuarios y asocie al usuario.

```
aws elasticache create-user-group \
  --user-group-id iam-user-group-01 \
  --engine redis \
  --user-ids default iam-user-01

aws elasticache modify-serverless-cache \
  --serverless-cache-name cache-01  \
  --user-group-id iam-user-group-01
```

Conexión

Conectar con el token como contraseña

Primero debe generar el token de autenticación de IAM de corta duración mediante una solicitud prefirmada SigV4 de AWS. A continuación, debe proporcionar el token de autenticación de IAM como contraseña al conectarse a una caché de Valkey o Redis OSS, como se muestra en el ejemplo siguiente.

```
String userId = "insert user id";
String cacheName = "insert cache name";
boolean isServerless = true;
String region = "insert region";

// Create a default AWS Credentials provider.
```

```
// This will look for AWS credentials defined in environment variables or system
 properties.
AWSCredentialsProvider awsCredentialsProvider = new
 DefaultAWSCredentialsProviderChain();

// Create an IAM authentication token request and signed it using the AWS credentials.
// The pre-signed request URL is used as an IAM authentication token for ElastiCache
 with Redis OSS.
IAMAuthTokenRequest iamAuthTokenRequest = new IAMAuthTokenRequest(userId, cacheName,
 region, isServerless);
String iamAuthToken =
 iamAuthTokenRequest.toSignedRequestUri(awsCredentialsProvider.getCredentials());

// Construct Redis OSS URL with IAM Auth credentials provider
RedisURI redisURI = RedisURI.builder()
    .withHost(host)
    .withPort(port)
    .withSsl(ssl)
    .withAuthentication(userId, iamAuthToken)
    .build();

// Create a new Lettuce Redis OSS client
RedisClient client = RedisClient.create(redisURI);
client.connect();
```

A continuación, se muestra la definición de `IAMAuthTokenRequest`.

```
public class IAMAuthTokenRequest {
    private static final HttpMethodName REQUEST_METHOD = HttpMethodName.GET;
    private static final String REQUEST_PROTOCOL = "http://";
    private static final String PARAM_ACTION = "Action";
    private static final String PARAM_USER = "User";
    private static final String PARAM_RESOURCE_TYPE = "ResourceType";
    private static final String RESOURCE_TYPE_SERVERLESS_CACHE = "ServerlessCache";
    private static final String ACTION_NAME = "connect";
    private static final String SERVICE_NAME = "elasticache";
    private static final long TOKEN_EXPIRY_SECONDS = 900;

    private final String userId;
    private final String cacheName;
    private final String region;
    private final boolean isServerless;
```

```
    public IAMAuthTokenRequest(String userId, String cacheName, String region, boolean
isServerless) {
        this.userId = userId;
        this.cacheName = cacheName;
        this.region = region;
        this.isServerless = isServerless;
    }

    public String toSignedRequestUri(AWSCredentials credentials) throws
URISyntaxException {
        Request<Void> request = getSignableRequest();
        sign(request, credentials);
        return new URIBuilder(request.getEndpoint())
            .addParameters(toNamedValuePair(request.getParameters()))
            .build()
            .toString()
            .replace(REQUEST_PROTOCOL, "");
    }

    private <T> Request<T> getSignableRequest() {
        Request<T> request  = new DefaultRequest<>(SERVICE_NAME);
        request.setHttpMethod(REQUEST_METHOD);
        request.setEndpoint(getRequestUri());
        request.addParameters(PARAM_ACTION, Collections.singletonList(ACTION_NAME));
        request.addParameters(PARAM_USER, Collections.singletonList(userId));
        if (isServerless) {
            request.addParameters(PARAM_RESOURCE_TYPE,
Collections.singletonList(RESOURCE_TYPE_SERVERLESS_CACHE));
        }
        return request;
    }

    private URI getRequestUri() {
        return URI.create(String.format("%s%s/", REQUEST_PROTOCOL, cacheName));
    }

    private <T> void sign(SignableRequest<T> request, AWSCredentials credentials) {
        AWS4Signer signer = new AWS4Signer();
        signer.setRegionName(region);
        signer.setServiceName(SERVICE_NAME);

        DateTime dateTime = DateTime.now();
        dateTime = dateTime.plus(Duration.standardSeconds(TOKEN_EXPIRY_SECONDS));
```

```
        signer.presignRequest(request, credentials, dateTime.toDate());
    }

    private static List<NameValuePair> toNamedValuePair(Map<String, List<String>> in) {
        return in.entrySet().stream()
            .map(e -> new BasicNameValuePair(e.getKey(), e.getValue().get(0)))
            .collect(Collectors.toList());
    }
}
```

Conectar con el proveedor de credenciales

En el siguiente código, se muestra cómo autenticarse con ElastiCache mediante el proveedor de credenciales de autenticación de IAM.

```
String userId = "insert user id";
String cacheName = "insert cache name";
boolean isServerless = true;
String region = "insert region";

// Create a default AWS Credentials provider.
// This will look for AWS credentials defined in environment variables or system
 properties.
AWSCredentialsProvider awsCredentialsProvider = new
 DefaultAWSCredentialsProviderChain();

// Create an IAM authentication token request. Once this request is signed it can be
 used as an
// IAM authentication token for ElastiCache with Redis OSS.
IAMAuthTokenRequest iamAuthTokenRequest = new IAMAuthTokenRequest(userId, cacheName,
 region, isServerless);

// Create a Redis OSS credentials provider using IAM credentials.
RedisCredentialsProvider redisCredentialsProvider = new
 RedisIAMAuthCredentialsProvider(
    userId, iamAuthTokenRequest, awsCredentialsProvider);

// Construct Redis OSS URL with IAM Auth credentials provider
RedisURI redisURI = RedisURI.builder()
    .withHost(host)
    .withPort(port)
    .withSsl(ssl)
    .withAuthentication(redisCredentialsProvider)
```

```
    .build();

// Create a new Lettuce Redis OSS client
RedisClient client = RedisClient.create(redisURI);
client.connect();
```

Más abajo, vemos un ejemplo de un cliente de Lettuce Redis OSS que incluye IAMAuthTokenRequest en un proveedor de credenciales para generar automáticamente credenciales temporales cuando sea necesario.

```
public class RedisIAMAuthCredentialsProvider implements RedisCredentialsProvider {
    private static final long TOKEN_EXPIRY_SECONDS = 900;

    private final AWSCredentialsProvider awsCredentialsProvider;
    private final String userId;
    private final IAMAuthTokenRequest iamAuthTokenRequest;
    private final Supplier<String> iamAuthTokenSupplier;

    public RedisIAMAuthCredentialsProvider(String userId,
        IAMAuthTokenRequest iamAuthTokenRequest,
        AWSCredentialsProvider awsCredentialsProvider) {
        this.userName = userName;
        this.awsCredentialsProvider = awsCredentialsProvider;
        this.iamAuthTokenRequest = iamAuthTokenRequest;
        this.iamAuthTokenSupplier =
 Suppliers.memoizeWithExpiration(this::getIamAuthToken, TOKEN_EXPIRY_SECONDS,
 TimeUnit.SECONDS);
    }

    @Override
    public Mono<RedisCredentials> resolveCredentials() {
        return Mono.just(RedisCredentials.just(userId, iamAuthTokenSupplier.get()));
    }

    private String getIamAuthToken() {
        return
 iamAuthTokenRequest.toSignedRequestUri(awsCredentialsProvider.getCredentials());
    }
}
```

Autenticación con el comando AUTH de Valkey o Redis OSS

> **Note**
>
> AUTH se ha reemplazado por the section called “Control de acceso basado en roles (RBAC)”. Todas las cachés sin servidor deben usar RBAC para la autenticación.

Las contraseñas o tokens de autenticación de Valkey y Redis OSS permiten a Valkey y Redis OSS exigir una contraseña para que los clientes puedan ejecutar comandos, lo que mejora la seguridad de los datos. AUTH está disponible solo para clústeres basados en nodos.

Temas

- Información general de AUTH en ElastiCache para Valkey y Redis OSS
- Aplicación de la autenticación a un clúster de ElastiCache para Valkey y Redis OSS
- Modificación del token de AUTH en un clúster existente
- Migración de RBAC a AUTH

Información general de AUTH en ElastiCache para Valkey y Redis OSS

Cuando se utiliza AUTH con el clúster de ElastiCache para Valkey y Redis OSS, existen algunas peculiaridades.

En concreto, tenga en cuenta estas restricciones de los tokens o contraseñas de AUTH al utilizar AUTH:

- Los tokens o contraseñas, deben tener entre 16 y 128 caracteres imprimibles.
- Los caracteres no alfanuméricos están restringidos a (!, &, #, $, ^, <, >, -).
- Solo se puede activar AUTH para los clústeres de Valkey o Redis OSS con el cifrado en tránsito activado.

Para configurar un token seguro, recomendamos que siga una política de contraseña estricta, como las que exigen las siguientes condiciones:

- Los tokens o contraseñas deben incluir al menos tres de los siguientes tipos de caracteres:
  - Caracteres en mayúsculas

  - Caracteres en minúsculas
  - Dígitos
  - Caracteres no alfanuméricos (!, &, #, $, ^, <, >, -)
- Los tokens o contraseñas no deben contener una palabra del diccionario ni una palabra del diccionario modificada ligeramente.
- Los tokens o contraseñas no deben ser iguales ni similares a un token que se haya utilizado recientemente.

Aplicación de la autenticación a un clúster de ElastiCache para Valkey y Redis OSS

Puede exigir a los usuarios que introduzcan un token (contraseña) en un servidor de Valkey o Redis OSS protegido mediante token. Para ello, incluya el parámetro `--auth-token` (API: `AuthToken`) con el token correcto al crear el clúster o grupo de reproducción. Inclúyalo también en todos los comandos posteriores para el clúster o grupo de reproducción.

La siguiente operación de AWS CLI crea un grupo de reproducción con el cifrado en tránsito (TLS) habilitado y el token de AUTH *`This-is-a-sample-token`*. Reemplace el grupo de subredes `sng-test` por otro existente.

Parámetros clave

- **`--engine`**: debe ser `valkey` o `redis`.
- **`--engine-version`**: si el motor es Redis OSS, debe ser la versión 3.2.6, 4.0.10 o una posterior.
- **`--transit-encryption-enabled`**: obligatorio para la autenticación y elegibilidad para la HIPAA.
- **`--auth-token`**: obligatorio para la conformidad con la HIPAA. Este valor deber ser el token correcto de este servidor de Valkey o Redis OSS protegido mediante token.
- **`--cache-subnet-group`**: obligatorio para la conformidad con la HIPAA.

Para Linux, macOS o Unix:

```
aws elasticache create-replication-group \
    --replication-group-id authtestgroup \
    --replication-group-description authtest \
    --engine redis \
    --cache-node-type cache.m4.large \
```

```
    --num-node-groups 1 \
    --replicas-per-node-group 2 \
    --transit-encryption-enabled \
    --auth-token This-is-a-sample-token \
    --cache-subnet-group sng-test
```

Para Windows:

```
aws elasticache create-replication-group ^
    --replication-group-id authtestgroup ^
    --replication-group-description authtest ^
    --engine redis ^
    --cache-node-type cache.m4.large ^
    --num-node-groups 1 ^
    --replicas-per-node-group 2 ^
    --transit-encryption-enabled ^
    --auth-token This-is-a-sample-token ^
    --cache-subnet-group sng-test
```

Modificación del token de AUTH en un clúster existente

Para facilitar la actualización de la autenticación, puede modificar el token de AUTH utilizado en un clúster. Puede realizar esta modificación si la versión del motor es Valkey 7.2 o una posterior o Redis 5.0.6 o una posterior. ElastiCache también debe tener habilitado el cifrado en tránsito.

La modificación del token de AUTH admite dos estrategias: ROTATE y SET. La estrategia ROTATE añade un token de AUTH adicional al servidor a la vez que se retiene el token anterior. La estrategia SET actualiza el servidor para que admita un único token de AUTH. Realice estas llamadas de modificación con el parámetro `--apply-immediately` para aplicar los cambios de inmediato.

Rotación del token AUTH

Para actualizar un servidor de Valkey o Redis OSS con un nuevo token de AUTH, llame a la API de `ModifyReplicationGroup` con el parámetro `--auth-token` como nuevo token de AUTH y `--auth-token-update-strategy` con el valor ROTATE. Una vez que finaliza la modificación de ROTATE, el clúster admitirá el token de AUTH anterior además del especificado en el parámetro `auth-token`. Si no se configuró ningún token de AUTH en el grupo de replicación antes de la rotación del token de AUTH, el clúster admite el token de AUTH especificado en el parámetro `--auth-token`, además de la conexión sin autenticación. Consulte Configuración del token AUTH para actualizar el token de AUTH que se va a exigir mediante la estrategia de actualización SET.

> **Note**
>
> Si no configura el token AUTH antes, una vez completada la modificación, el clúster no admitirá ningún token AUTH además del especificado en el parámetro auth-token.

Si esta modificación se realiza en un servidor que ya admite dos tokens de AUTH, el token de AUTH más antiguo también se eliminará durante esta operación. Esto permite que un servidor admita hasta los dos token de AUTH más recientes en un momento determinado.

En este momento, puede continuar con la actualización del cliente para que utilice el último token de AUTH. Una vez que se actualizan los clientes, puede utilizar la estrategia SET para la rotación del token AUTH (que se explica en la sección siguiente) a fin de comenzar a utilizar en exclusiva el token nuevo.

La siguiente operación de la AWS CLI modifica un grupo de reproducción para rotar el token de AUTH *This-is-the-rotated-token*.

Para Linux, macOS o Unix:

```
aws elasticache modify-replication-group \
--replication-group-id authtestgroup \
--auth-token This-is-the-rotated-token \
--auth-token-update-strategy ROTATE \
--apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group ^
--replication-group-id authtestgroup ^
--auth-token This-is-the-rotated-token ^
--auth-token-update-strategy ROTATE ^
--apply-immediately
```

Configuración del token AUTH

Para actualizar un servidor de Valkey o Redis OSS para que admita un único token de AUTH obligatorio, llame a la operación de la API `ModifyReplicationGroup` con el parámetro `--auth-token` con el mismo valor que el último token de AUTH y el parámetro `--auth-token-update-strategy` con el valor `SET`. La estrategia SET solo se puede usar con un clúster que tenga dos

token de AUTH o un token de AUTH opcional debido al uso anterior de una estrategia ROTATE. Una vez finalizada la modificación, el servidor solo admite el token de AUTH especificado en el parámetro auth-token.

La siguiente operación de la AWS CLI modifica un grupo de replicación para configurar el token de AUTH en *This-is-the-set-token*.

Para Linux, macOS o Unix:

```
aws elasticache modify-replication-group \
--replication-group-id authtestgroup \
--auth-token This-is-the-set-token \
--auth-token-update-strategy SET \
--apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group ^
--replication-group-id authtestgroup ^
--auth-token This-is-the-set-token ^
--auth-token-update-strategy SET ^
--apply-immediately
```

Habilitación de la autenticación en un clúster existente

Para habilitar la autenticación en un servidor de Valkey o Redis OSS existente, llame a la operación de la API `ModifyReplicationGroup`. Llame a `ModifyReplicationGroup` con el parámetro `--auth-token` como el nuevo token y `--auth-token-update-strategy` con el valor ROTATE.

Tras finalizar la modificación de ROTATE, el clúster admite el token de AUTH especificado en el parámetro `--auth-token` además de la conexión sin autenticación. Una vez que todas las aplicaciones cliente estén actualizadas para autenticarse en Valkey o Redis OSS con el token de AUTH, utilice la estrategia SET para marcar el token de AUTH como obligatorio. La habilitación de la autenticación solo se admite en servidores de Valkey y Redis OSS con el cifrado en tránsito (TLS) habilitado.

Migración de RBAC a AUTH

Si está autenticando usuarios con el control de acceso basado en roles (RBAC) de Valkey o Redis OSS, tal y como se describe en Control de acceso basado en roles (RBAC), y desea migrar a AUTH, utilice los siguientes procedimientos. Puede migrar mediante la consola o la CLI.

Migración desde RBAC a AUTH mediante la consola

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En la lista de la esquina superior derecha, elija la región de AWS en la que se encuentra el clúster que desea modificar.
3. En el panel de navegación, elija el motor que se ejecuta en el clúster que desea modificar.

   Se mostrará una lista de los clústeres del motor elegido.
4. En la lista de clústeres, para el clúster que desea modificar, elija su nombre.
5. Para Actions (Acciones), elija Modify (Modificar).

   Aparecerá la ventana Modificar.
6. En Control de acceso, elija Acceso de usuario predeterminado de AUTH de Valkey o Acceso de usuario predeterminado de AUTH de Redis OSS.
7. En Token de AUTH de Valkey o Token de AUTH de Redis OSS, defina un nuevo token.
8. Seleccione Vista previa de los cambios y seleccione Modificar en la siguiente pantalla.

Migración desde RBAC a AUTH mediante la AWS CLI

Utilice uno de los siguientes comandos para configurar un nuevo token de AUTH opcional para su grupo de replicación de Valkey o Redis OSS. Tenga en cuenta que un token de autenticación opcional permitirá el acceso no autenticado al grupo de replicación hasta que el token de autenticación se marque como obligatorio, de acuerdo con la estrategia de actualización SET que se describe en el siguiente paso.

Para Linux, macOS o Unix:

```
aws elasticache modify-replication-group \
    --replication-group-id test \
    --remove-user-groups \
    --auth-token This-is-a-sample-token \
    --auth-token-update-strategy ROTATE \
    --apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group ^
```

```
    --replication-group-id test ^
    --remove-user-groups ^
    --auth-token This-is-a-sample-token ^
    --auth-token-update-strategy ROTATE ^
    --apply-immediately
```

Tras ejecutar el comando anterior, puede actualizar sus aplicaciones de Valkey o Redis OSS para que realicen la autenticación en el grupo de replicación de ElastiCache mediante el token de AUTH opcional recién configurado. Para completar la rotación del token de autenticación, utilice la estrategia de actualización `SET` en el siguiente comando. De esta forma, el token de AUTH opcional se marcará como obligatorio. Cuando se complete la actualización del token de autenticación, el estado del grupo de replicación aparecerá como `ACTIVE` y todas las conexiones a este grupo de replicación requerirán autenticación.

Para Linux, macOS o Unix:

```
aws elasticache modify-replication-group \
   --replication-group-id test \
   --auth-token This-is-a-sample-token \
   --auth-token-update-strategy SET \
   --apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group ^
   --replication-group-id test ^
   --remove-user-groups ^
   --auth-token This-is-a-sample-token ^
   --auth-token-update-strategy SET ^
   --apply-immediately
```

Para obtener más información, consulte Autenticación con el comando AUTH de Valkey o Redis OSS.

> **Note**
>
> Si necesita deshabilitar el control de acceso en un clúster de ElastiCache, consulte the section called “Deshabilitación del control de acceso en una caché de ElastiCache Valkey o Redis OSS”.

Deshabilitación del control de acceso en una caché de ElastiCache Valkey o Redis OSS

Siga las instrucciones que se indican a continuación para deshabilitar el control de acceso en una caché de Valkey o Redis OSS habilitada para TLS. Su caché tendrá uno de los siguientes dos tipos de configuraciones: acceso de usuario predeterminado de AUTH o lista de control de acceso de grupos de usuarios (RBAC). Si la caché se creó con la configuración AUTH, debe cambiarla a la configuración RBAC antes de poder deshabilitar la caché mediante la eliminación de los grupos de usuarios. Si la caché se creó con la configuración RBAC, puede proceder directamente a deshabilitarla.

Deshabilitación de una caché sin servidor de Valkey o Redis OSS configurada con RBAC

1. Elimine los grupos de usuarios para deshabilitar el control de acceso.

```
aws elasticache modify-serverless-cache --serverless-cache-name <serverless-cache>
 --remove-user-group
```

2. (Opcional) Compruebe que no haya ningún grupo de usuarios asociado a la caché sin servidor.

```
aws elasticache describe-serverless-caches --serverless-cache-name <serverless-
cache>
{
    "..."
    "UserGroupId": ""
    "..."
}
```

Deshabilitación de una caché de Valkey o Redis OSS configurada con un token de AUTH

1. Cambie el token AUTH a RBAC y especifique un grupo de usuarios que agregar.

```
aws elasticache modify-replication-group --replication-group-id <replication-group-
id-value> --auth-token-update-strategy DELETE --user-group-ids-to-add <user-group-
value>
```

2. Compruebe que el token AUTH se haya desactivado y que se haya agregado un grupo de usuarios.

```
aws elasticache describe-replication-groups --replication-group-id <replication-
group-id-value>
```

```
{
    "..."
    "AuthTokenEnabled": false,
    "UserGroupIds": [
        "<user-group-value>"
    ]
    "..."
}
```

3. Elimine los grupos de usuarios para deshabilitar el control de acceso.

```
aws elasticache modify-replication-group --replication-group-id <replication-group-
value> --user-group-ids-to-remove <user-group-value>
{
    "..."
    "PendingModifiedValues": {
    "UserGroups": {
      "UserGroupIdsToAdd": [],
      "UserGroupIdsToRemove": [
        "<user-group-value>"
      ]
    }
    "..."
}
```

4. (Opcional) Compruebe que no haya ningún grupo de usuarios asociado al clúster. Además, el campo `AuthTokenEnabled` debe decir false.

```
aws elasticache describe-replication-groups --replication-group-id <replication-
group-value>
"AuthTokenEnabled": false
```

Deshabilitación de un clúster de Valkey o Redis OSS configurado con RBAC

1. Elimine los grupos de usuarios para deshabilitar el control de acceso.

```
aws elasticache modify-replication-group --replication-group-id <replication-group-
value> --user-group-ids-to-remove <user-group-value>
{
    "..."
    "PendingModifiedValues": {
    "UserGroups": {
```

```
      "UserGroupIdsToAdd": [],
      "UserGroupIdsToRemove": [
        "<user-group-value>"
      ]
    }
    "..."
}
```

2. (Opcional) Compruebe que no haya ningún grupo de usuarios asociado al clúster. Además, el campo `AuthTokenEnabled` debe decir false.

```
aws elasticache describe-replication-groups --replication-group-id <replication-
group-value>
"AuthTokenEnabled": false
```

Privacidad del tráfico entre redes

Amazon ElastiCache utiliza las siguientes técnicas para proteger los datos de su caché frente a accesos no autorizados:

- Seguridad de ElastiCache y Amazon VPC explica el tipo de grupo de seguridad que necesita para su instalación.
- Identity and Access Management para Amazon ElastiCache para conceder y limitar las acciones de los usuarios, grupos y roles.

Temas

- Seguridad de ElastiCache y Amazon VPC
- Puntos de conexión de VPC de interfaz y API de ElastiCache (AWS PrivateLink)
- Subredes y grupos de subredes

Seguridad de ElastiCache y Amazon VPC

Puesto que la seguridad de los datos es importante, ElastiCache ofrece medios para que pueda controlar quién tiene acceso a los datos. El modo en que controla el acceso a sus datos depende de si ha lanzado o no sus clústeres en una Amazon Virtual Private Cloud (Amazon VPC) o en Amazon EC2-Classic.

> **Important**
>
> Hemos dado de baja la utilización de Amazon EC2-Classic para lanzar clústeres de ElastiCache. Todos los nodos de la generación actual se lanzan exclusivamente en Amazon Virtual Private Cloud.

El servicio de Amazon Virtual Private Cloud (Amazon VPC) define una red virtual que se parece mucho a un centro de datos tradicional. Al configurar su Amazon VPC, puede seleccionar su rango de direcciones IP, crear subredes y configurar las tablas de enrutamiento, las gateways de red y la configuración de seguridad. También puede agregar un clúster a la red virtual y controlar el acceso al clúster mediante los grupos de seguridad de Amazon VPC.

En esta sección se explica cómo configurar un clúster de ElastiCache manualmente en una Amazon VPC. Esta información se ha pensado para usuarios que desean tener conocimientos más amplios acerca de cómo funcionan ElastiCache y Amazon VPC de manera conjunta.

Temas

- Comprensión de ElastiCache y Amazon VPC
- Patrones de acceso para obtener acceso a una caché de ElastiCache en una Amazon VPC
- Creación de una Virtual Private Cloud (VPC)
- Conexión a una caché que se ejecuta en una Amazon VPC

Comprensión de ElastiCache y Amazon VPC

ElastiCache se encuentra totalmente integrado con la Amazon Virtual Private Cloud (Amazon VPC). Para los usuarios de ElastiCache, esto significa lo siguiente:

- Si su cuenta de AWS solo admite la plataforma EC2-VPC, ElastiCache siempre lanza el clúster en una Amazon VPC.
- Si es la primera vez que utiliza AWS, los clústeres se implementarán en una Amazon VPC. Para ello, se creará una VPC predeterminada automáticamente.
- Si tiene una VPC predeterminada y no especifica una subred al lanzar un clúster, el clúster se lanzará en su Amazon VPC predeterminada.

Para obtener más información, consulte Detección de plataformas compatibles y comprobación de si tiene una VPC predeterminada.

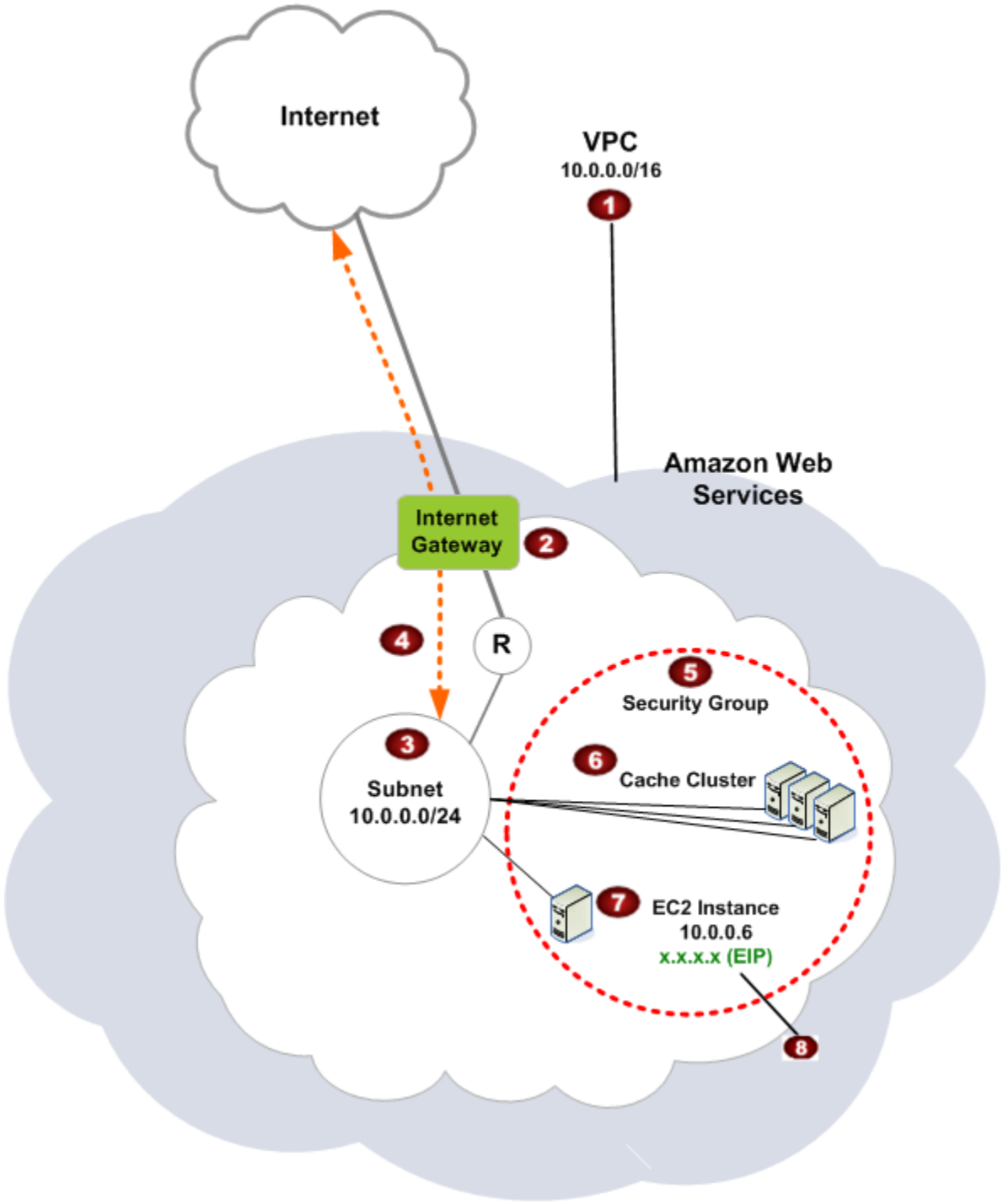
Con Amazon Virtual Private Cloud, podrá crear una red virtual en la nube de AWS que se parece mucho a un centro de datos tradicional. Puede configurar su Amazon VPC, incluido seleccionar el rango de direcciones IP, crear subredes y configurar las tablas de enrutamiento, las gateways de red y la configuración de seguridad.

La funcionalidad básica de ElastiCache funciona exactamente igual en una nube virtual privada; ElastiCache administra las actualizaciones de software, la aplicación de parches, la detección de errores y la recuperación, tanto si los clústeres se han implementado dentro o fuera de una Amazon VPC.

Los nodos de caché de ElastiCache implementados fuera de una Amazon VPC tienen asignada una dirección IP en la que se resuelve el punto de conexión/nombre de DNS. Esto proporciona conectividad de instancias de Amazon Elastic Compute Cloud (Amazon EC2). Al lanzar un clúster de ElastiCache en una subred privada de Amazon VPC, se asigna a todos los nodos de caché una dirección IP privada dentro de dicha subred.

Información general sobre ElastiCache en una Amazon VPC

A continuación, se proporcionan un diagrama y una tabla que describen el entorno de Amazon VPC, junto con los clústeres de ElastiCache y las instancias de Amazon EC2 que se lanzaron en la Amazon VPC.



1	La Amazon VPC es una parte aislada de la nube de AWS a la que se asigna su propio bloque de direcciones IP.
2	

	Una gateway de Internet conecta la Amazon VPC directamente a Internet y proporciona acceso a otros recursos de AWS como Amazon Simple Storage Service (Amazon S3) que se ejecutan fuera de su Amazon VPC.
3	Una subred de Amazon VPC es un segmento del rango de direcciones IP de una Amazon VPC donde podrá aislar recursos de AWS en función de sus necesidades operativas y de seguridad.
4	Una tabla de enrutamiento en la Amazon VPC dirige el tráfico de red entre la subred e Internet. La Amazon VPC tiene un enrutador implícito, representado en este diagrama mediante el círculo que contiene la letra R.
5	Un grupo de seguridad de Amazon VPC controla el tráfico entrante y saliente de sus clústeres de ElastiCache y sus instancias de Amazon EC2.
6	Puede lanzar un clúster de ElastiCache en la subred. Los nodos de caché tienen direccion es IP privadas del rango de direcciones de la subred.
7	También puede lanzar instancias de Amazon EC2 en la subred. Cada instancia de Amazon EC2 tiene una dirección IP privada del rango de direcciones de la subred. La instancia de Amazon EC2 puede conectarse a cualquier nodo de caché de la misma subred.
8	Para que se pueda obtener acceso a una instancia de Amazon EC2 de su Amazon VPC desde Internet, deberá asignar a la instancia una dirección pública y estática denominada dirección IP elástica.

Requisitos previos

Para crear un clúster de ElastiCache en una Amazon VPC, esta Amazon VPC debe cumplir los siguientes requisitos:

- La Amazon VPC debe permitir instancias de Amazon EC2 no dedicadas. No puede utilizar ElastiCache en una Amazon VPC configurada para la tenencia de instancias dedicadas.

- Debe definir un grupo de subredes de caché para su Amazon VPC. ElastiCache utiliza dicho grupo de subredes de caché para seleccionar una subred y direcciones IP pertenecientes a ella y asociarlas a sus puntos de conexión de VPC o nodos de caché.
- Los bloques de CIDR de cada subred deben ser lo suficientemente grandes como para proporcionar direcciones IP auxiliares para que pueda utilizarlas durante las actividades de mantenimiento.

Enrutamiento y seguridad

Puede configurar el enrutamiento en la Amazon VPC para controlar dónde fluye el tráfico (por ejemplo, a la gateway de Internet o la gateway privada virtual). Con una gateway de Internet, la Amazon VPC tiene acceso directo a otros recursos de AWS que no se ejecutan en su Amazon VPC. Si decide tener solo una gateway privada virtual con una conexión a la red local de su organización, puede enrutar el tráfico vinculado a Internet a través de la VPN y utilizar políticas de seguridad locales y firewalls para controlar las salidas. En ese caso, se cobrarán tarifas de banda ancha adicionales al obtener acceso a los recursos de AWS a través de Internet.

Puede utilizar grupos de seguridad de Amazon VPC para ayudar a proteger los clústeres de ElastiCache y las instancias de Amazon EC2 de la Amazon VPC. Los grupos de seguridad actúan como un firewall en el ámbito de la instancia, no en el de la subred.

> **Note**
>
> Recomendamos utilizar nombres de DNS para conectarse a los nodos de caché, ya que la dirección IP subyacente puede cambiar.

Documentación de Amazon VPC

Amazon VPC tiene su propia serie de documentación que describe cómo crear y utilizar una Amazon VPC. En la siguiente tabla, se proporcionan enlaces a las guías de Amazon VPC.

Descripción	Documentación
Cómo comenzar a utilizar Amazon VPC	Introducción a Amazon VPC
Cómo utilizar Amazon VPC a través de la Consola de administración de AWS	Guía del usuario de Amazon VPC

Descripción	Documentación
Descripciones completas de todos los comandos de Amazon VPC	Referencia de línea de comandos de Amazon EC2 (los comandos de Amazon VPC se encuentran en la referencia de Amazon EC2)
Descripciones completas de las operaciones, los tipos de datos y los errores de la API de Amazon VPC	Referencia de la línea de comandos de Amazon EC2 (las operaciones de la API de Amazon VPC se encuentran en la referencia de Amazon EC2)
Información para el administrador de red que necesita configurar la gateway en su extremo para disponer de una conexión de VPN IPsec opcional	¿Qué es AWS Site-to-Site VPN?

Para obtener información más detallada sobre Amazon Virtual Private Cloud, consulte Amazon Virtual Private Cloud.

Patrones de acceso para obtener acceso a una caché de ElastiCache en una Amazon VPC

Amazon ElastiCache admite los siguientes escenarios para obtener acceso a una caché en una Amazon VPC:

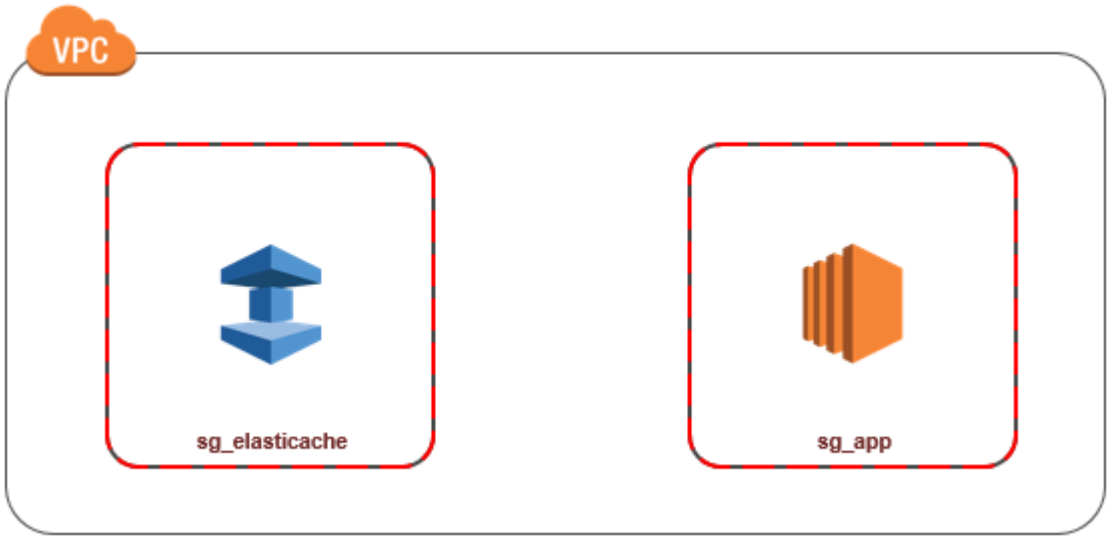
Contenido

- Acceso a una caché de ElastiCache cuando esta y la instancia de Amazon EC2 se encuentran en la misma Amazon VPC
- Acceso a una caché de ElastiCache cuando esta y la instancia de Amazon EC2 se encuentran en Amazon VPC diferentes
  - Acceso a una caché de ElastiCache cuando esta y la instancia de Amazon EC2 se encuentran en Amazon VPC diferentes en la misma región
    - Uso de Transit Gateway
  - Acceso a una caché de ElastiCache cuando esta y la instancia de Amazon EC2 se encuentran en Amazon VPC diferentes en regiones distintas
    - Uso de la VPC de tránsito
- Acceso a una caché de ElastiCache desde una aplicación en ejecución en un centro de datos del cliente
  - Acceso a una caché de ElastiCache desde una aplicación en ejecución en un centro de datos del cliente mediante conectividad de VPN
  - Acceso a una caché de ElastiCache desde una aplicación en ejecución en un centro de datos del cliente mediante Direct Connect

Acceso a una caché de ElastiCache cuando esta y la instancia de Amazon EC2 se encuentran en la misma Amazon VPC

El caso de uso más común es cuando una aplicación implementada en una instancia EC2 debe conectarse a una caché en la misma VPC.

En el siguiente diagrama se ilustra este escenario.



La forma más sencilla de administrar el acceso entre instancias EC2 y cachés de la misma VPC es hacer lo siguiente:

1. Crear un grupo de seguridad VPC para la caché. Este grupo de seguridad se puede utilizar para restringir el acceso a la caché. Por ejemplo, puede crear una regla personalizada para este grupo de seguridad que permita el acceso mediante TCP utilizando el puerto que asignó a la caché cuando la creó y una dirección IP que se utilizará para obtener acceso a la caché.

   El puerto predeterminado para las cachés de Memcached es 11211.

   El puerto predeterminado para cachés de Valkey y Redis OSS es 6379.
2. Cree un grupo de seguridad de VPC para sus instancias EC2 (servidores web y de aplicaciones). Si es necesario, este grupo de seguridad puede permitir el acceso a la instancia EC2 desde Internet a través de la tabla de enrutamiento de la VPC. Por ejemplo, puede establecer reglas en este grupo de seguridad para permitir el acceso mediante TCP a la instancia EC2 a través del puerto 22.
3. Crear reglas personalizadas en el grupo de seguridad para la caché que permitan las conexiones desde el grupo de seguridad que ha creado para las instancias EC2. Esto permitirá a cualquier miembro del grupo de seguridad obtener acceso a las cachés.

> **Note**
>
> Si planea utilizar Local Zones, asegúrese de que las ha habilitado. Cuando crea un grupo de subredes en esa zona local, la VPC se amplía a dicha zona local y la VPC tratará la subred

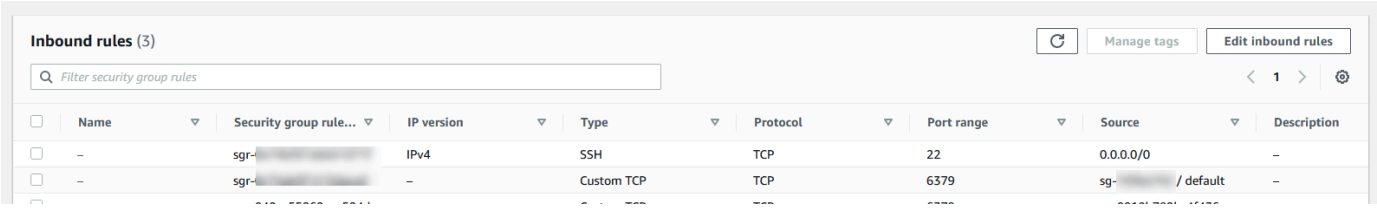
como cualquier subred de cualquier otra zona de disponibilidad. Todas las gateways y tablas de enrutamiento relevantes se ajustarán de forma automática.

Para crear una regla en un grupo de seguridad de VPC que permita establecer conexiones desde otro grupo de seguridad

1. Inicie sesión en la consola de administración de AWS y abra la consola de Amazon VPC en https://console.aws.amazon.com/vpc.
2. En el panel de navegación, elija Grupos de seguridad.
3. Seleccione o cree un grupo de seguridad que utilizará para la caché En Inbound Rules (Reglas de entrada), seleccione Edit Inbound Rules (Editar reglas de entrada) y, a continuación, seleccione Add Rule (Agregar regla). Este grupo de seguridad permitirá el acceso a los miembros de otro grupo de seguridad.
4. En Type (Tipo), elija Custom TCP Rule (Personalizar regla de TCP).
   a. En Rango de puerto, especifique el puerto que utilizó al crear la caché.

      El puerto predeterminado para las cachés de Memcached es 11211.

      El puerto predeterminado para las cachés y los grupos de replicación de Valkey y Redis OSS es 6379.
   b. En el cuadro Source (Fuente), comience a escribir el ID del grupo de seguridad. Desde la lista, seleccione el grupo de seguridad que utilizará para sus instancias de Amazon EC2.
5. Cuando haya terminado, elija Save (Guardar).



Acceso a una caché de ElastiCache cuando esta y la instancia de Amazon EC2 se encuentran en Amazon VPC diferentes

Cuando una caché está en una VPC que no coincide con la de la instancia EC2 que se está utilizando para acceder a ella, hay varias formas de obtener acceso a la caché. Si la caché y la instancia EC2 están en VPC distintas, pero en la misma región, puede utilizar la conexión de

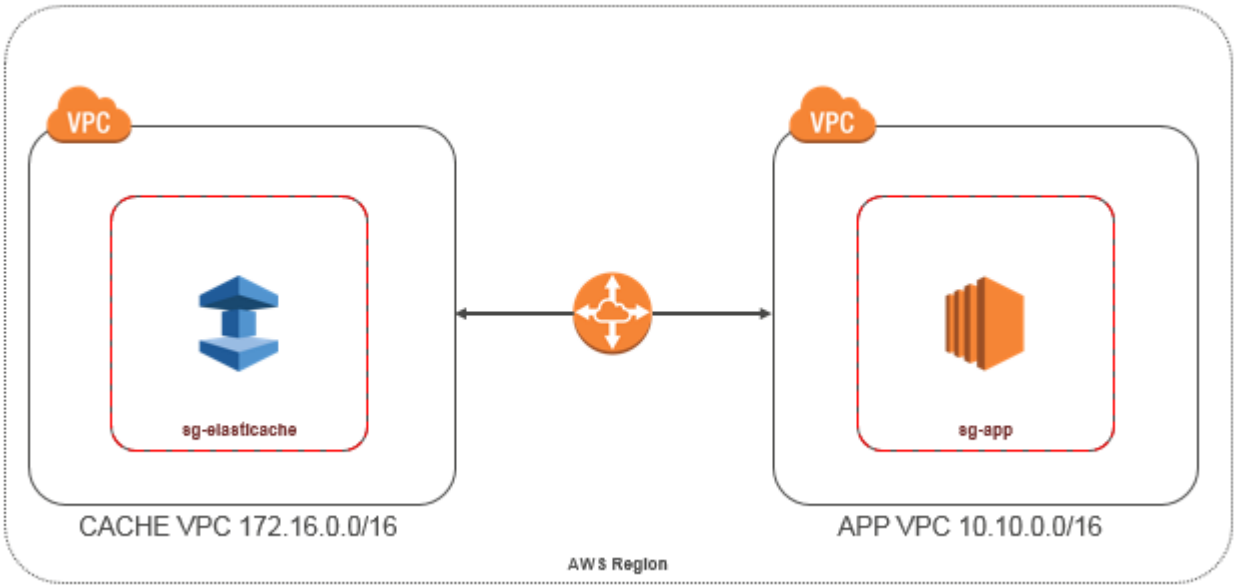
emparejamiento de VPC. Si la caché y la instancia EC2 se encuentran en distintas regiones, puede crear conectividad de VPN entre regiones.

Temas

- Acceso a una caché de ElastiCache cuando esta y la instancia de Amazon EC2 se encuentran en Amazon VPC diferentes en la misma región
- Acceso a una caché de ElastiCache cuando esta y la instancia de Amazon EC2 se encuentran en Amazon VPC diferentes en regiones distintas

Acceso a una caché de ElastiCache cuando esta y la instancia de Amazon EC2 se encuentran en Amazon VPC diferentes en la misma región

En el siguiente diagrama, se ilustra el acceso a una caché por parte de una instancia de Amazon EC2 en una Amazon VPC diferente y en la misma región a través de una conexión de emparejamiento de VPC.



Acceso a caché de una instancia de Amazon EC2 en una Amazon VPC diferente dentro de la misma región: conexión de emparejamiento de VPC

Una conexión de emparejamiento de VPC es una conexión de redes entre dos VPC que permite direccionar el tráfico entre ellas mediante direcciones IP privadas. Las instancias de ambas VPC se pueden comunicar entre sí siempre que se encuentren en la misma red. Puede crear una

conexión de emparejamiento de VPC entre sus propias Amazon VPC o con una Amazon VPC de otra cuenta de AWS dentro de una misma región. Para obtener más información sobre la conexión de emparejamiento de Amazon VPC, consulte la documentación de VPC.

> **Note**
>
> La resolución de nombres DNS puede fallar para las VPC emparejadas, según las configuraciones aplicadas a la VPC de ElastiCache. Para solucionar esto, ambas VPC deben tener habilitados los nombres de host DNS y la resolución de DNS. Para obtener más información, consulte Habilitación de la resolución de DNS para la conexión de emparejamiento de VPC.

Para obtener acceso a una caché en una Amazon VPC diferente a través de emparejamiento

1. Asegúrese de que las dos VPC no tengan rangos de IP solapados o no podrá interconectarlas.
2. Coloque las dos VPC al mismo nivel. Para obtener más información, consulte Creación y aceptación de interconexiones de Amazon VPC.
3. Actualice su tabla de ruteo. Para obtener más información, consulte Actualización de las tablas de ruteo para interconexiones de VPC

   A continuación, se muestra el aspecto que tienen las tablas de ruteo para el ejemplo del diagrama anterior. Tenga en cuenta que pcx-a894f1c1 es la conexión de emparejamiento.

Destination	Target
172.16.0.0/16	local
10.10.0.0/16	pcx-a894f1c1

Destination	Target
10.10.0.0/16	local
0.0.0.0/0	igw-bfdcccd8
172.16.0.0/16	pcx-a894f1c1

   Tabla de enrutamiento de VPC
4. Modifique el grupo de seguridad de su caché de ElastiCache para permitir la conexión de entrada del grupo de seguridad de la aplicación en la VPC del mismo nivel. Para obtener más información, consulte Actualización de los grupos de seguridad para que hagan referencia a grupos de la VPC del mismo nivel.

El acceso a una caché a través de una conexión de emparejamiento generará costos de transferencia de datos adicionales.

Uso de Transit Gateway

Una gateway de tránsito permite adjuntar las VPC y las conexiones de VPN de la misma región de AWS y enrutar el tráfico entre ellas. Una gateway de tránsito funciona en todas las cuentas de AWS y puede utilizar AWS Resource Access Manager para compartir la gateway de tránsito con otras cuentas. Después de compartir una transit gateway con otra cuenta de AWS, el propietario de la cuenta puede asociar las VPC a la gateway de tránsito. Un usuario de cualquiera de las cuentas puede eliminar la vinculación en cualquier momento.

Puede habilitar la multidifusión en una gateway de tránsito y, a continuación, crear un dominio de multidifusión de transit gateway que permita que el trafico de multidifusión se envíe desde el origen de multidifusión a los miembros del grupo de multidifusión a través de conexiones de la VPC que asocie con el dominio.

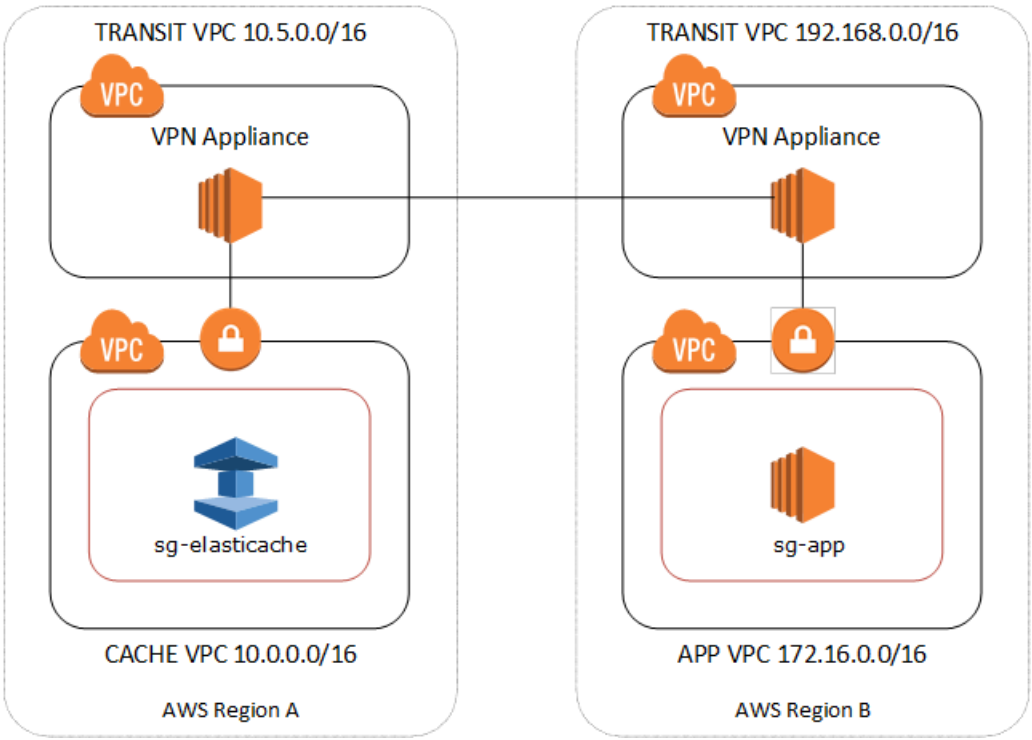
También puede crear una conexión de emparejamiento entre transit gateways de diferentes regiones de AWS. Esto le permite dirigir el tráfico entre las vinculaciones de las transit gateways a través de diferentes regiones.

Para obtener más información, consulte Transit gateways.

Acceso a una caché de ElastiCache cuando esta y la instancia de Amazon EC2 se encuentran en Amazon VPC diferentes en regiones distintas

Uso de la VPC de tránsito

Una alternativa a la utilización de la conexión de emparejamiento de VPC, otra estrategia común para conectar varias VPC y redes remotas dispersas geográficamente es crear una VPC de tránsito que sirva como un centro de tránsito de red global. Una VPC de tránsito simplifica la administración de la red y minimiza el número de conexiones necesarias para conectar varias VPC y redes remotas. Este diseño puede ahorrar tiempo y esfuerzo, además de reducir los costos, ya que se implementa prácticamente sin los gastos tradicionales de establecer una presencia física en un hub de tránsito de coubicación o de implementar un equipo de red física.



Conexión entre VPC diferentes en distintas regiones

Una vez que la Amazon VPC de tránsito se encuentre establecida, una aplicación implementada en una VPC “radial” en una región se puede conectar a una caché de ElastiCache en una VPC “radial” dentro de otra región.

Para obtener acceso a una caché en una VPC diferente dentro de una región de AWS distinta

1. Implemente una solución de VPC de tránsito. Para obtener más información, consulte AWS Transit Gateway.
2. Actualice las tablas de enrutamiento de la VPC en la aplicación y las VPC de la caché para direccionar el tráfico a través de la VGW (gateway privada virtual) y el dispositivo de VPN. En caso de que se produzca el enrutamiento dinámico con el protocolo de gateway fronteriza (BGP), las rutas se pueden propagar automáticamente.
3. Modifique el grupo de seguridad de su caché de ElastiCache para permitir la conexión de entrada del rango de IP de instancias de aplicación. Tenga en cuenta que no podrá remitirse al grupo de seguridad de servidor de la aplicación en este caso.

El acceso a una caché entre regiones conllevará latencias de red y costos adicionales de transferencia de datos entre regiones.

Acceso a una caché de ElastiCache desde una aplicación en ejecución en un centro de datos del cliente

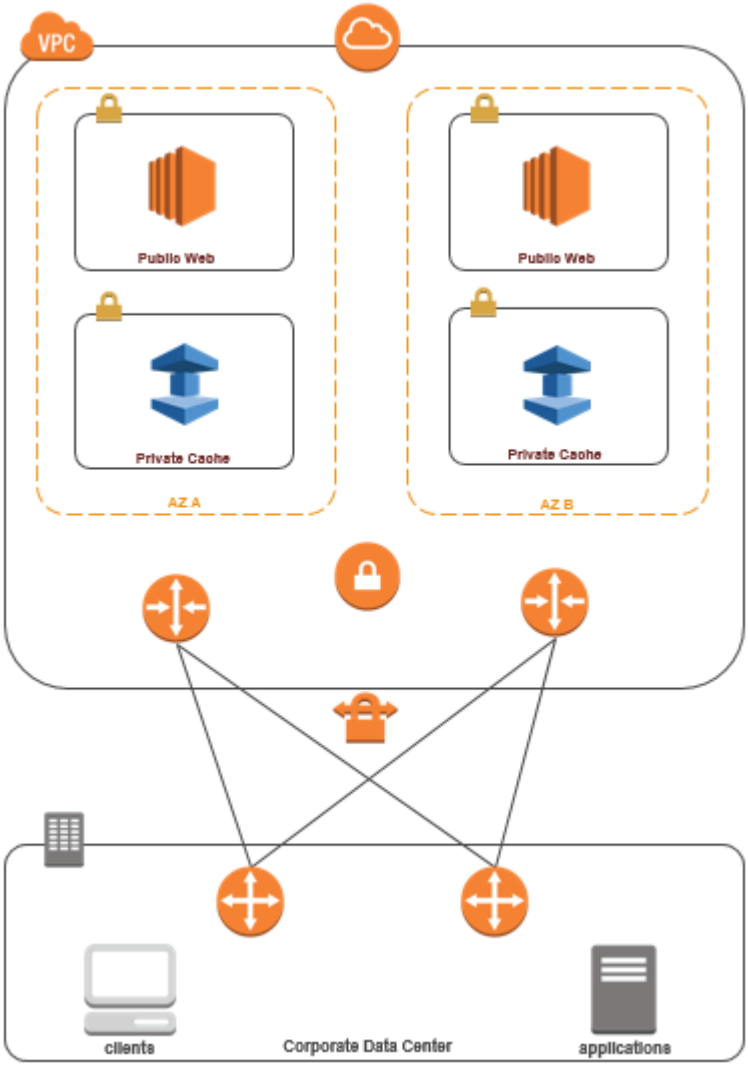
Otra situación posible es una arquitectura híbrida en la que los clientes o las aplicaciones del centro de datos del cliente puedan necesitar acceso a una caché de ElastiCache en la VPC. Esta situación también se admite, siempre que haya conectividad entre la VPC del cliente y el centro de datos, ya sea a través de la VPN como de Direct Connect.

Temas

- Acceso a una caché de ElastiCache desde una aplicación en ejecución en un centro de datos del cliente mediante conectividad de VPN
- Acceso a una caché de ElastiCache desde una aplicación en ejecución en un centro de datos del cliente mediante Direct Connect

Acceso a una caché de ElastiCache desde una aplicación en ejecución en un centro de datos del cliente mediante conectividad de VPN

En el siguiente diagrama, se muestra el acceso a una caché de ElastiCache desde una aplicación que se ejecuta en la red de su empresa a través de conexiones de VPN.



Conexión a ElastiCache desde su centro de datos a través de una VPN

Para obtener acceso a una caché en una VPC desde una aplicación local a través de una conexión de VPN

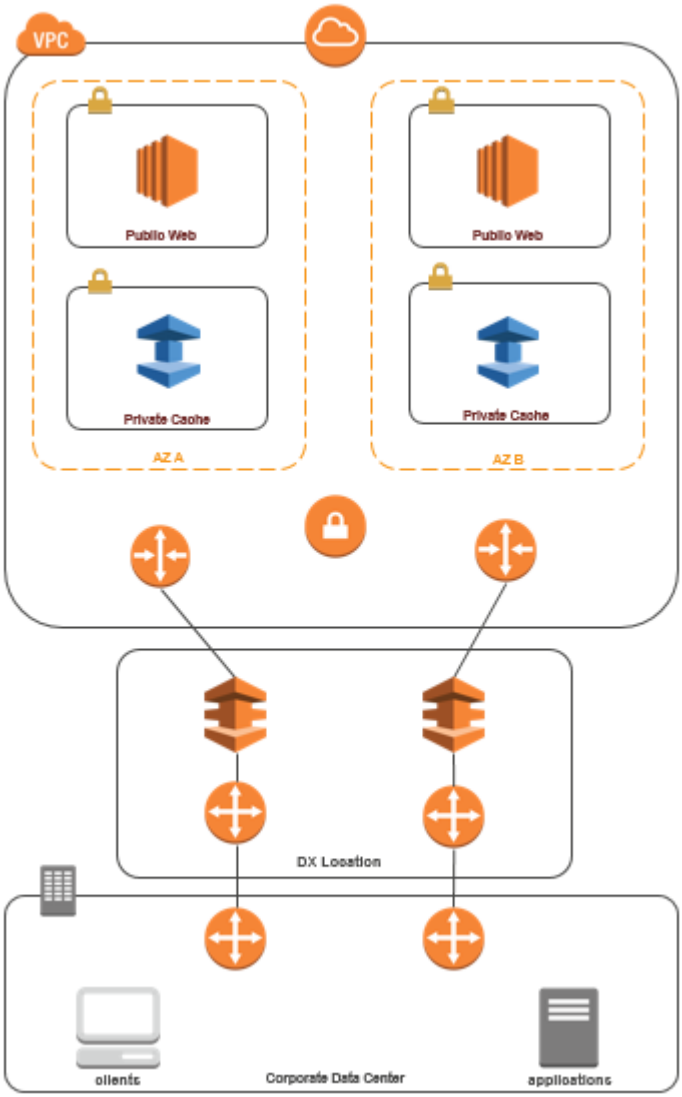
1. Para establecer la conectividad de VPN, agregue una gateway privada virtual de hardware a su VPC. Para obtener más información, consulte Adición de una gateway privada virtual de hardware a la VPC.
2. Actualice la tabla de enrutamiento de VPC para la subred en la que se implementa su caché de ElastiCache a fin de permitir el tráfico desde el servidor de aplicaciones en las instalaciones. En caso de que se produzca el enrutamiento dinámico con BGP, las rutas se pueden propagar automáticamente.

3. Modifique el grupo de seguridad de su caché de ElastiCache para permitir la conexión de entrada desde los servidores de aplicaciones en las instalaciones.

El acceso a una caché a través de una conexión de VPN conllevará latencias de red y costos adicionales de transferencia de datos.

Acceso a una caché de ElastiCache desde una aplicación en ejecución en un centro de datos del cliente mediante Direct Connect

En el siguiente diagrama, se muestra el acceso a una caché de ElastiCache desde una aplicación que se ejecuta en la red de su empresa a través de Direct Connect.



Conexión a ElastiCache desde su centro de datos a través de Direct Connect

Para obtener acceso a una caché de ElastiCache desde una aplicación en ejecución en su red mediante Direct Connect

1. Establezca la conectividad de Direct Connect. Para obtener más información, consulte Introducción a AWS Direct Connect.
2. Modifique el grupo de seguridad de su caché de ElastiCache para permitir la conexión de entrada desde los servidores de aplicaciones en las instalaciones.

El acceso a una caché a través de una conexión de DX puede conllevar latencias de red y cargos adicionales por transferencia de datos.

Creación de una Virtual Private Cloud (VPC)

En este ejemplo, creará una Amazon VPC con una subred privada para cada zona de disponibilidad.

Creación de una Amazon VPC (consola)

1. Inicie sesión en la consola de administración de AWS y abra la consola de Amazon VPC en https://console.aws.amazon.com/vpc/.
2. En el panel de VPC, elija Create VPC (Crear VPC).
3. En Recursos para crear elija VPC y más.
4. En Number of Availability Zones (AZs) (Número de zonas de disponibilidad), seleccione el número de zonas de disponibilidad que quiere usar con las subredes.
5. En Number of public subnets (Número de subredes públicas), elija el número de subredes públicas que desea agregar a la VPC.
6. En Number of private subnets (Número de subredes privadas), elija el número de subredes públicas que desea agregar a la VPC.

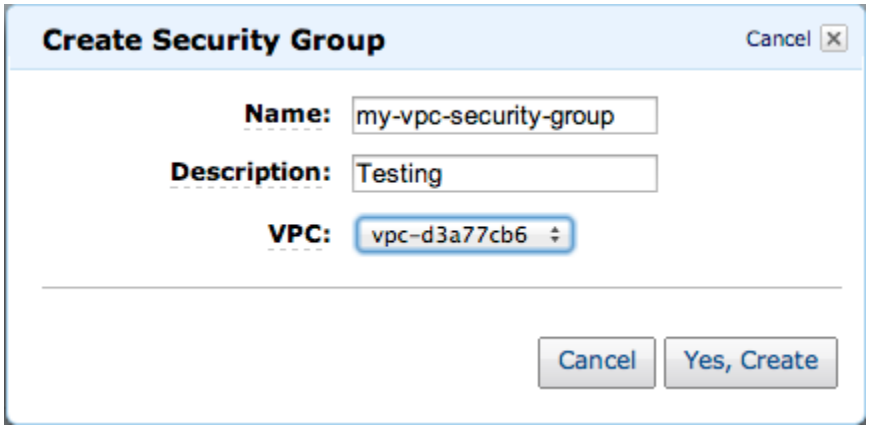
   > **Tip**
   >
   > Anote los identificadores de las subredes e identifique cuál es pública y cuál es privada. Necesitará esta información más adelante al lanzar sus clústeres y agregar una instancia de Amazon EC2 a su Amazon VPC.

7. Cree un grupo de seguridad de Amazon VPC. Utilizará este grupo para su clúster y su instancia de Amazon EC2.

   a. En el panel de navegación de la consola de administración de Amazon VPC, elija Security Groups (Grupos de seguridad).

   b. Elija Creación de grupo de seguridad.

   c. Escriba un nombre y una descripción para el grupo de seguridad en los cuadros correspondientes. En el cuadro VPC, elija el identificador de su Amazon VPC.



d. Una vez que la configuración sea la deseada, elija Yes, Create (Sí, crear).

8. Defina una regla de entrada de red para su grupo de seguridad. Esta regla permitirá conectarse a su instancia de Amazon EC2 mediante Secure Shell (SSH).

   a. En la lista de navegación, elija Security Groups (Grupos de seguridad).

   b. Busque el grupo de seguridad en la lista y, a continuación, elíjalo.

   c. En Security Group (Grupo de seguridad), elija la pestaña Inbound (Entrada). En el cuadro Create a new rule (Crear una nueva regla), elija SSH y, a continuación, elija Add Rule (Agregar regla).

   d. Establezca los siguientes valores para la regla de entrada nueva a fin de permitir el acceso HTTP:

      - Tipo: HTTP
      - Fuente: 0.0.0.0/0

      Elija Apply Rule Changes (Aplicar cambios de regla).

Ahora ya puede crear un grupo de subredes de caché y lanzar un clúster en su Amazon VPC.

- Creación de un grupo de subredes
- Creación de un clúster de Memcached (consola).
- Creación de un clúster de Valkey (modo de clúster deshabilitado) (consola).

Conexión a una caché que se ejecuta en una Amazon VPC

En este ejemplo, se muestra cómo lanzar una instancia de Amazon EC2 en su Amazon VPC. A continuación, podrá iniciar sesión en esta instancia y obtener acceso a la caché de ElastiCache que se ejecuta en la Amazon VPC.

Conexión a una caché que se ejecuta en una Amazon VPC (consola)

En este ejemplo, crea una instancia de Amazon EC2 en su Amazon VPC. Puede utilizar esta instancia de Amazon EC2 para conectarse a nodos de caché que se ejecutan en la Amazon VPC.

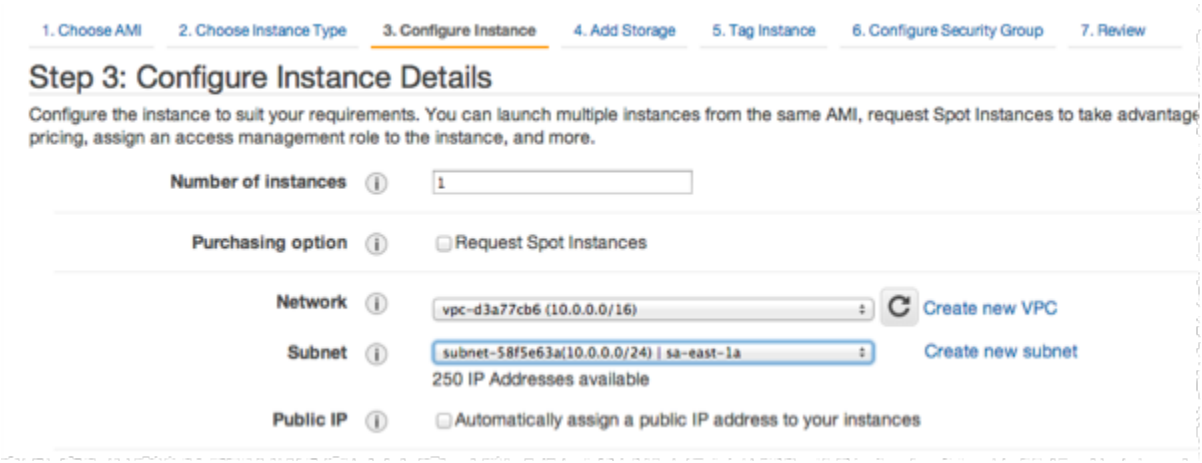
> **Note**
>
> Para obtener información sobre la utilización de Amazon EC2, consulte la Guía de introducción a Amazon EC2 en la Documentación de Amazon EC2.

Para crear una instancia de Amazon EC2 en su Amazon VPC mediante la consola de Amazon EC2

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon EC2 en https://console.aws.amazon.com/ec2/.
2. En la consola, elija Launch Instance (Lanzar instancia) y siga los pasos siguientes:
3. En la página Choose an Amazon Machine Image (AMI) (Elegir una Amazon Machine Image [AMI]), elija una AMI de Amazon Linux de 64 bits y, a continuación, elija Select (Seleccionar).
4. En la página de Choose an Instance Type (Elegir un tipo de instancias), elija 3. Configure Instance (Configurar instancia.
5. En la página Configure Instance Details (Configurar detalles de instancia), seleccione lo siguiente:

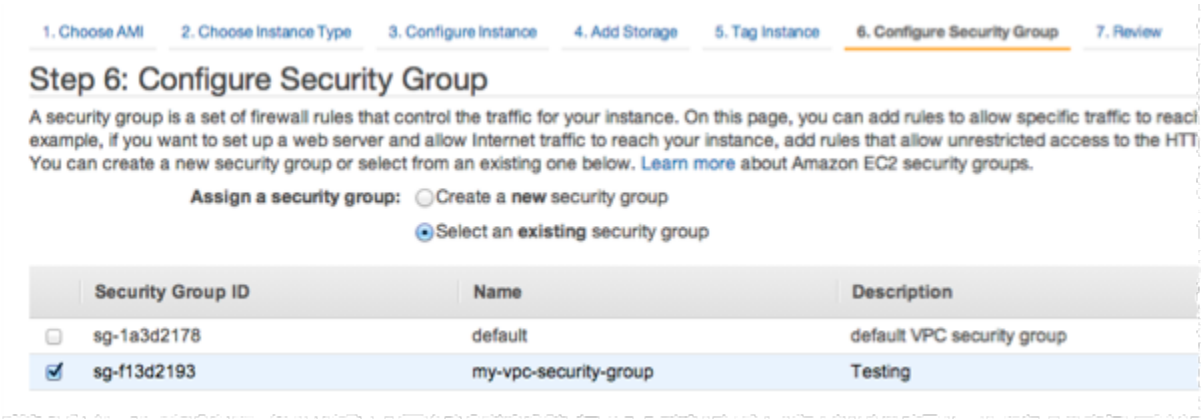
    a. En la lista de Network (Red), elija su Amazon VPC.

    b. En la lista Subred, elija su subred pública.



Cuando esté conforme con los ajustes, elija 4. Add Storage (Agregar almacenamiento.

6. En la página de Add Storage (Agregar almacenamiento), elija 5. Tag Instance (Etiquetar instancia.
7. En la página de Tag Instance (Etiquetar instancia), escriba un nombre para su instancia de Amazon EC2 y, a continuación, elija 6. Configure Security Group (Configurar grupo de seguridad.
8. En la página Configure Security Group (Configurar grupo de seguridad), elija Select an existing security group (Seleccionar un grupo de seguridad existente). Para obtener más información sobre los grupos de seguridad, consulte Grupos de seguridad de Amazon EC2 para instancias Linux.



Elija el nombre de su grupo de seguridad de Amazon VPC y, a continuación, elija Review and Launch (Revisar y lanzar).

9. En la página Review Instance and Launch (Revisar instancia y lanzar), elija Launch (Lanzar).

10. En la ventana Select an existing key pair or create a new key pair (Seleccionar un par de claves existente o crear uno nuevo), especifique el par de claves que desea utilizar con esta instancia.

    > **Note**
    >
    > Para obtener información sobre la administración de pares de claves, consulte la Guía de introducción a Amazon EC2.

11. Cuando se encuentre listo para lanzar la instancia de Amazon EC2, elija Launch (Lanzar).

Ahora puede asignar una dirección IP elástica a la instancia de Amazon EC2 que acaba de crear. Necesitará utilizar esta dirección IP para conectarse a la instancia de Amazon EC2.

Para asignar una dirección IP elástica (consola)

1. Abra la consola de Amazon VPC en https://console.aws.amazon.com/vpc/.
2. En la lista de navegación, elija Elastic IPs (IP elásticas).
3. Elija Asignar dirección IP elástica.
4. En el cuadro de diálogo Allocate Elastic IP address (Asignar dirección IP elástica) acepte el Network Border Group (Grupo de bordes de red) predeterminado y elija Allocate (Asignar).
5. Elija la dirección IP elástica que acaba de asignar de la lista y elija Associate Address (Asociar dirección).
6. En el cuadro de diálogo de Associate Address (Asociar dirección), en el cuadro de Instance (Instancia), elija el ID de la instancia de Amazon EC2 que ha lanzado.

   En el cuadro Private IP address (Dirección IP privada), seleccione el cuadro para obtener la dirección IP privada y, a continuación, elija Associate (Asociar).

   Ahora puede utilizar SSH para conectarse a la instancia de Amazon EC2 mediante la dirección IP elástica que ha creado.

Para conectarse con la instancia de Amazon EC2, siga estos pasos:

- Abra una ventana del sistema. En el símbolo del sistema, envíe el comando siguiente y reemplace mykeypair.pem por el nombre de su archivo de par de claves y 54.207.55.251 con su dirección IP elástica.

```
ssh -i mykeypair.pem ec2-user@54.207.55.251
```

> **Important**
>
> Todavía no cierre la sesión de su instancia de Amazon EC2.

Ahora se encuentra preparado para interactuar con su clúster de ElastiCache. Para ello, debe instalar la utilidad telnet, si es que todavía no lo ha hecho.

Cómo instalar telnet e interactuar con su clúster (AWS CLI)

- Abra una ventana del sistema. En el símbolo del sistema, envíe el comando que se indica a continuación. En el mensaje de confirmación, escriba y.

```
sudo yum install telnet
Loaded plugins: priorities, security, update-motd, upgrade-helper
Setting up Install Process
Resolving Dependencies
--> Running transaction check

...(output omitted)...

Total download size: 63 k
Installed size: 109 k
Is this ok [y/N]: y
Downloading Packages:
telnet-0.17-47.7.amzn1.x86_64.rpm                          |  63 kB     00:00

...(output omitted)...

Complete!
```

Ahora puede conectarse a una VPC con Memcached o Redis.

Conexión a una VPC con Memcached

1. Diríjase a la consola de ElastiCache en https://console.aws.amazon.com/elasticache/ y obtenga el punto de conexión de uno de los nodos de su clúster. Para obtener más información, consulte Búsqueda de puntos de enlace de conexión.
2. Utilice la utilidad telnet para conectarse a su punto de conexión de nodo de caché a través del puerto 11211. Reemplace el nombre de host que se muestra a continuación con el nombre de host de su nodo de caché.

   ```
   telnet my-cache-cluster.7wufxa.0001.use1.cache.amazonaws.com 11211
   ```

   Ya está conectado a la caché del motor y puede enviar comandos. En este ejemplo, se añade un elemento de datos a la caché para, a continuación, obtenerlo inmediatamente. Por último, desconéctese del nodo de caché.

   Para almacenar una clave y un valor, escriba las dos líneas siguientes:

   ```
   add mykey 0 3600 28
   This is the value for mykey
   ```

   El motor de caché responde con lo siguiente:

   ```
   OK
   ```

   Para recuperar el valor de `mykey`, escriba lo siguiente:

   ```
   get mykey
   ```

   El motor de caché responde con lo siguiente:

   ```
   VALUE mykey 0 28
   This is the value for my key
   END
   ```

   Para desconectarse del motor de caché, escriba lo siguiente:

   ```
   quit
   ```

Conexión a una VPC con Redis

1. Diríjase a la consola de ElastiCache en https://console.aws.amazon.com/elasticache/ y obtenga el punto de conexión de uno de los nodos de su clúster. Para obtener más información, consulte Finding connection endpoints para Redis.
2. Utilice telnet para conectarse a su punto de conexión de nodo de caché a través del puerto 6379. Reemplace el nombre de host que se muestra a continuación con el nombre de host de su nodo de caché.

   ```
   telnet my-cache-cluster.7wufxa.0001.use1.cache.amazonaws.com 6379
   ```

   Ya está conectado a la caché del motor y puede enviar comandos. En este ejemplo, se añade un elemento de datos a la caché para, a continuación, obtenerlo inmediatamente. Por último, desconéctese del nodo de caché.

   Para almacenar una clave y un valor, escriba las dos líneas siguientes:

   ```
   set mykey myvalue
   ```

   El motor de caché responde con lo siguiente:

   ```
   OK
   ```

   Para recuperar el valor de `mykey`, escriba lo siguiente:

   ```
   get mykey
   ```

   Para desconectarse del motor de caché, escriba lo siguiente:

   ```
   quit
   ```

3. Diríjase a la consola de ElastiCache en https://console.aws.amazon.com/elasticache/ y obtenga el punto de conexión de uno de los nodos de su clúster. Para obtener más información, consulte Finding connection endpoints para Redis OSS.
4. Utilice telnet para conectarse a su punto de conexión de nodo de caché a través del puerto 6379. Reemplace el nombre de host que se muestra a continuación con el nombre de host de su nodo de caché.

```
telnet my-cache-cluster.7wufxa.0001.use1.cache.amazonaws.com 6379
```

Ya está conectado a la caché del motor y puede enviar comandos. En este ejemplo, se añade un elemento de datos a la caché para, a continuación, obtenerlo inmediatamente. Por último, desconéctese del nodo de caché.

Para almacenar una clave y un valor, escriba lo siguiente:

```
set mykey myvalue
```

El motor de caché responde con lo siguiente:

```
OK
```

Para recuperar el valor de `mykey`, escriba lo siguiente:

```
get mykey
```

El motor de caché responde con lo siguiente:

```
get mykey
myvalue
```

Para desconectarse del motor de caché, escriba lo siguiente:

```
quit
```

> **Important**
>
> Para evitar incurrir en cargos adicionales en su cuenta de AWS, asegúrese de eliminar los recursos de AWS que ya no desea después de probar estos ejemplos.

Puntos de conexión de VPC de interfaz y API de ElastiCache (AWS PrivateLink)

Puede establecer una conexión privada entre los puntos de conexión de la VPC y la API de Amazon ElastiCache mediante la creación de un punto de conexión de VPC de interfaz . Los puntos de conexión de interfaz que cuentan con AWS PrivateLink. AWS PrivateLink le permite acceder de forma privada a las operaciones de la API de Amazon ElastiCache sin una puerta de enlace de Internet, un dispositivo NAT, una conexión VPN o una conexión AWS Direct Connect.

Las instancias de la VPC no necesitan direcciones IP públicas para comunicarse con el punto de conexión de la API de Amazon ElastiCache. Sus instancias tampoco necesitan direcciones IP públicas para utilizar ninguna de las operaciones de la API ElastiCache de RDS disponibles. El tráfico entre su VPC y Amazon ElastiCache no sale de la red de Amazon. Cada punto de conexión de la interfaz está representado por una o más interfaces de red elásticas en las subredes. Para obtener más información sobre las interfaces de red elásticas, consulte Interfaces de red elásticas en la Guía del usuario de Amazon EC2.

- Para obtener más información sobre puntos de conexión de la VPC, consulte Puntos de enlace de la VPC de tipo interfaz (AWS PrivateLink) en la Guía del usuario de Amazon VPC.
- Para obtener más información sobre las operaciones de la API de ElastiCache, consulte Operaciones de API de ElastiCache.

Después de crear un punto de conexión de VPC de tipo interfaz, si habilita nombres de host DNS privados para el punto de conexión, el punto de conexión predeterminado de ElastiCache https://elasticache.*Region*.amazonaws.com) se resuelve en el punto de conexión de VPC. Si no habilita nombres de host de DNS privados, Amazon VPC proporciona un nombre de punto de conexión de DNS que puede utilizar en el siguiente formato:

```
VPC_Endpoint_ID.elasticache.Region.vpce.amazonaws.com
```

Para obtener más información, consulte Puntos de conexión de la VPC de tipo interfaz (AWS PrivateLink) en la Guía del usuario de Amazon VPC. ElastiCache permite realizar llamadas a todas sus acciones de API dentro de su VPC.

> **Note**
>
> Los nombres de host DNS privados solo se pueden habilitar para un punto de conexión de VPC en la VPC. Si quiere crear un punto de conexión de VPC adicional, el nombre de host DNS privado debe estar deshabilitado para ese punto de conexión.

Regiones de Privatelink disponibles

Código	Ubicación	Región
CPT	África (Ciudad del Cabo)	AF-SOUTH-1
HKG	Asia-Pacífico (Hong Kong)	AP-EAST-1
TPE	Asia-Pacífico (Taipéi)	AP-EAST-2
NRT	Asia-Pacífico (Tokio)	AP-NORTHEAST-1
ICN	Asia-Pacífico (Seúl)	AP-NORTHEAST-2
KIX	Asia Pacífico (Osaka)	AP-NORTHEAST-3
BOM (lista de materiales)	Asia-Pacífico (Bombay)	AP-SOUTH-1
HYD	Asia-Pacífico (Hyderabad)	AP-SOUTH-2
SIN	Asia-Pacífico (Singapur)	AP-SOUTHEAST-1
SYD	Asia-Pacífico (Sídney)	AP-SOUTHEAST-2
CGK	Asia-Pacífico (Yakarta)	AP-SOUTHEAST-3
MEL	Asia-Pacífico (Melbourne)	AP-SOUTHEAST-4
KUL	Asia-Pacífico (Malasia)	AP-SOUTHEAST-5
BKK	Asia-Pacífico (Tailandia)	AP-SOUTHEAST-7
YUL	Canadá (centro)	CA-CENTRAL-1
YYC	Oeste de Canadá (Calgary)	CA-WEST-1

Código	Ubicación	Región
BJS	China (Pekín)	CN-NORTH-1
ZHY	China (Ningxia)	CN-NORTHWEST-1
FRA	Europa (Fráncfort)	EU-CENTRAL-1
ZRH	Europa (Zúrich)	EU-CENTRAL-2
ARN	Europa (Estocolmo)	EU-NORTH-1
MXP	Europa (Milán)	EU-SOUTH-1
ZAZ	Europa (España)	EU-SOUTH-2
DUB	Europa (Irlanda)	EU-WEST-1
LHR	Europa (Londres)	EU-WEST-2
CDG	Europa (París)	EU-WEST-3
TLV	Tel Aviv (Israel)	IL-CENTRAL-1
DXB	Medio Oriente (EAU)	ME-CENTRAL-1
BAH	Medio Oriente (Baréin)	ME-SOUTH-1
QRO	México (central)	MX-CENTRAL-1
GRU	América del Sur (São Paulo)	SA-EAST-1
IAD	Este de EE. UU. (Norte de Virginia)	US-EAST-1
CMH	Este de EE. UU. (Ohio)	US-EAST-2
OSU	AWS GovCloud (Este de EE. UU.)	US-GOV-EAST-1
SFO	Oeste de EE. UU. (Norte de California)	US-WEST-1

Código	Ubicación	Región
PDX	Oeste de EE. UU. (Oregón)	US-WEST-2
PDT	AWS GovCloud (Oeste de EE. UU.)	US-WEST-1

Consideraciones para los puntos de enlace de VPC

Antes de configurar un punto de conexión de VPC de tipo interfaz para los puntos de conexión de la API de Amazon ElastiCache, asegúrese de revisar Propiedades y limitaciones del puntos de conexión de la interfaz en la Guía del usuario de Amazon VPC. Todas las operaciones de API de ElastiCache relevantes para la administración de los recursos Amazon ElastiCache están disponibles desde la VPC mediante el uso de AWS PrivateLink.

Las políticas de puntos de conexión de VPC son compatibles con los puntos de conexión de API de ElastiCache. De forma predeterminada, se permite el acceso completo a las operaciones de API de ElastiCache a través del punto de conexión. Para obtener más información, consulte Control del acceso a los servicios con puntos de conexión de la VPC en la guía del usuario de Amazon VPC.

Creación de un punto de conexión de VPC de interfaz para la API de ElastiCache

Puede crear un punto de conexión de VPC para la API de Amazon ElastiCache mediante la consola de Amazon VPC o la AWS CLI. Para obtener más información, consulte Creación de un punto de conexión de interfaz en la Guía del usuario de Amazon VPC.

Después de crear un punto de conexión de VPC de interfaz, puede habilitar nombres de host de DNS privados para el punto de conexión. Cuando lo haga, el punto de conexión predeterminado de Amazon ElastiCache (https://elasticache.*Region*.amazonaws.com) se resuelve en el punto de conexión de VPC. En las regiones de China (Pekín) y China (Ningxia) de AWS, puede realizar solicitudes de la API con el punto de conexión de VPC mediante `elasticache.cn-north-1.amazonaws.com.cn` para Pekín y `elasticache.cn-northwest-1.amazonaws.com.cn` para Ningxia. Para obtener más información, consulte Acceso a un servicio a través de un punto de conexión de interfaz en la Guía del usuario de Amazon VPC.

Creación de una política de punto de conexión de VPC para la API de Amazon ElastiCache

Puede asociar una política de punto de conexión con el punto de conexión de VPC que controla el acceso a la API de ElastiCache. La política especifica lo siguiente:

- La entidad principal que puede realizar acciones.
- Las acciones que se pueden realizar.
- Los recursos en los que se pueden llevar a cabo las acciones.

Para obtener más información, consulte Control del acceso a los servicios con puntos de conexión de la VPC en la Guía del usuario de Amazon VPC.

Example Política de punto de conexión de VPC para acciones de la API de ElastiCache con Valkey o Redis OSS

A continuación, se muestra un ejemplo de una política de punto de conexión para la API de ElastiCache. Cuando se asocia a un punto de conexión, esta política concede acceso a las acciones de la API de ElastiCache enumeradas para todos las entidades principales de todos los recursos.

```
{
 "Statement": [{
  "Principal": "*",
  "Effect": "Allow",
  "Action": [
   "elasticache:CreateCacheCluster",
   "elasticache:ModifyCacheCluster",
   "elasticache:CreateSnapshot"
  ],
  "Resource": "*"
 }]
}
```

Example Política de punto de conexión de VPC para acciones de la API de ElastiCache para Memcached

A continuación, se muestra un ejemplo de una política de punto de conexión para la API de ElastiCache. Cuando se asocia a un punto de conexión, esta política concede acceso a las acciones de la API de ElastiCache enumeradas para todos las entidades principales de todos los recursos.

```
{
 "Statement": [{
  "Principal": "*",
  "Effect": "Allow",
  "Action": [
   "elasticache:CreateCacheCluster",
   "elasticache:ModifyCacheCluster"
  ],
  "Resource": "*"
 }]
}
```

Example Política de punto de conexión de VPC que deniega todo el acceso desde una cuenta de AWS especificada

La siguiente política de punto de conexión de VPC deniega a la cuenta de AWS *123456789012* todo el acceso a los recursos mediante el punto de conexión. La política permite todas las acciones de otras cuentas.

```
{
 "Statement": [{
   "Action": "*",
   "Effect": "Allow",
   "Resource": "*",
   "Principal": "*"
  },
  {
   "Action": "*",
   "Effect": "Deny",
   "Resource": "*",
   "Principal": {
    "AWS": [
     "123456789012"
    ]
   }
  }
 ]
}
```

Subredes y grupos de subredes

Un grupo de subredes es una colección de subredes (que suelen ser privadas) que puede designar para los clústeres basados en nodos que se ejecutan en un entorno de Amazon Virtual Private Cloud (VPC).

Si crea un clúster basado en nodos en una Amazon VPC, debe usar un grupo de subredes. ElastiCache utiliza dicho grupo de subredes para elegir una subred y direcciones IP pertenecientes a dicha subred a fin de asociarlas a sus nodos.

ElastiCache proporciona un grupo de subredes IPv4 predeterminado o puede optar por crear uno nuevo. Para IPv6, debe crear un grupo de subred con un bloque de CIDR IPv6. Si elige dual stack (doble pila), debe seleccionar un tipo de IP de detección, IPv6 o IPv4.

ElastiCache Serverless no utiliza un recurso de grupo de subredes, sino que toma una lista de subredes directamente durante la creación.

En esta sección, se explica cómo crear y aprovechar las subredes y los grupos de subredes para administrar el acceso a los recursos de ElastiCache.

Para obtener más información sobre la utilización de grupos de subredes en entornos de Amazon VPC, consulte Acceso al clúster o grupo de replicación de ElastiCache.

Temas

- Creación de un grupo de subredes
- Asignación de un grupo de subredes a una caché
- Modificación de un grupo de subredes
- Eliminación de un grupo de subredes

Creación de un grupo de subredes

Un grupo de subredes de caché es una colección de subredes que quizá quiera designar para sus cachés en una VPC. Al lanzar una caché en una VPC, tiene que seleccionar un grupo de subredes de caché. A continuación, ElastiCache utilizará dicho grupo de subredes de caché a fin de asignar direcciones IP de esta subred para cada nodo de caché en la caché.

Cuando cree un nuevo grupo de subredes, tenga en cuenta el número de direcciones IP disponibles. Si la subred tiene pocas direcciones IP libres, es posible que el número de nodos que pueda agregar a un clúster sea limitado. Para solucionar este problema, puede asignar una o varias subredes a un grupo de subredes para, de este modo, disponer de suficientes direcciones IP en la zona de disponibilidad de su clúster. Hecho esto, podrá agregar más nodos a su clúster.

Si elige IPV4 como tipo de red, estará disponible un grupo de subredes predeterminado o puede crear uno nuevo. ElastiCache utiliza dicho grupo de subredes para elegir una subred y direcciones IP pertenecientes a dicha subred a fin de asociarlas a sus nodos. Si elige IPV6 o doble pila, se le indicará que cree subredes IPV6 o de doble pila. Para obtener más información sobre los tipos de red, consulte Elección de un tipo de red en ElastiCache. Para obtener más información, consulte Create a subnet in your VPC (Crear una subred en la VPC).

En los siguientes procedimientos, se muestra cómo crear un grupo de subredes denominado `mysubnetgroup` (consola), la AWS CLI y la API de ElastiCache.

Creación de un grupo de subredes (consola)

En el siguiente procedimiento, se muestra cómo crear un grupo de subredes (consola).

Para crear un grupo de subredes (consola)

1. Inicie sesión en la AWS Management Console y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En la lista de navegación, elija Grupos de subredes.
3. Elija Create subnet group (Crear grupo de subredes).
4. En el asistente Crear grupo de subredes, haga lo siguiente. Cuando esté conforme con todos los ajustes, elija Crear.

    a. En el cuadro Name (Nombre), escriba un nombre para el grupo de subredes.
    b. En el cuadro Description(Descripción), escriba la descripción del grupo de subredes.
    c. En el cuadro ID de VPC, elija su Amazon VPC.

   d. Todas las subredes están seleccionadas de forma predeterminada. En el panel Subredes seleccionadas, haga clic en Administrar y seleccione las zonas de disponibilidad o las Zonas Locales y los ID de sus subredes privadas; luego, seleccione Elegir.

5. En el mensaje de confirmación que aparece, elija Close (Cerrar).

El grupo de subredes nuevo aparecerá en la lista Subnet Groups (Grupos de subredes) de la consola de ElastiCache. En la parte inferior de la ventana, podrá elegir el grupo de subredes para ver detalles tales como todas las subredes asociadas al grupo.

Creación de un grupo de subredes (AWS CLI)

En el símbolo del sistema, utilice el comando `create-cache-subnet-group` para crear un grupo de subredes.

Para Linux, macOS o Unix:

```
aws elasticache create-cache-subnet-group \
    --cache-subnet-group-name mysubnetgroup \
    --cache-subnet-group-description "Testing" \
    --subnet-ids subnet-53df9c3a
```

Para Windows:

```
aws elasticache create-cache-subnet-group ^
    --cache-subnet-group-name mysubnetgroup ^
    --cache-subnet-group-description "Testing" ^
    --subnet-ids subnet-53df9c3a
```

Este comando debería producir un resultado similar al siguiente:

```
{
    "CacheSubnetGroup": {
        "VpcId": "vpc-37c3cd17",
        "CacheSubnetGroupDescription": "Testing",
        "Subnets": [
            {
                "SubnetIdentifier": "subnet-53df9c3a",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2a"
                }
```

```
            }
        ],
        "CacheSubnetGroupName": "mysubnetgroup"
    }
}
```

Para obtener más información, consulte el tema AWS CLI de la create-cache-subnet-group.

Asignación de un grupo de subredes a una caché

Una vez que ha creado un grupo de subredes, puede lanzar una caché en una Amazon VPC. Para obtener más información, consulte lo siguiente.

- Clúster de Memcached: para lanzar un clúster de Memcached, consulte Creación de un clúster de Memcached (consola). En el paso 7.a (Configuración avanzada de Memcached), elija un grupo de subredes para la VPC.
- Clúster de Valkey o Redis OSS independiente: para iniciar un clúster de Valkey o Redis OSS de un solo nodo, consulte Creación de un clúster de Valkey (modo de clúster deshabilitado) (consola). En el paso 7.a (Configuración avanzada de Redis OSS), elija un grupo de subredes para la VPC.
- Grupo de replicación de Redis OSS (modo de clúster deshabilitado): para iniciar un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) en una VPC, consulte Creación de un grupo de replicación de Valkey o Redis OSS (modo de clúster deshabilitado) desde cero. En el paso 7.b (Configuración avanzada de Redis OSS), elija un grupo de subredes para la VPC.
- Grupo de replicación de Valkey o Redis OSS (modo de clúster habilitado): Creación de un clúster de Valkey o Redis OSS (modo de clúster habilitado) (consola). En el paso 6.i (Configuración avanzada de Redis OSS), elija un grupo de subredes para la VPC.

Modificación de un grupo de subredes

Puede modificar la descripción de un grupo de subredes o modificar la lista de los ID de subred asociados al grupo de subredes. No puede eliminar un ID de subred desde un grupo de subredes si hay una caché que utiliza actualmente dicha subred.

Los procedimientos que se describen a continuación muestran cómo modificar un grupo de subredes.

Modificación de grupos de subredes (consola)

Para modificar un grupo de subredes

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Subnet groups (grupos de subredes).
3. En la lista de grupos de subredes, seleccione el botón de opción del grupo que desee modificar y seleccione Modificar.
4. En el panel Subredes seleccionadas, elija Administrar.
5. Haga los cambios que desee en las subredes seleccionadas y haga clic en Elegir.
6. Haga clic en Guardar cambios para guardar los cambios.

Modificación de grupos de subredes (AWS CLI)

En el símbolo del sistema, use el comando `modify-cache-subnet-group` para modificar un grupo de subredes.

Para Linux, macOS o Unix:

```
aws elasticache modify-cache-subnet-group \
    --cache-subnet-group-name mysubnetgroup \
    --cache-subnet-group-description "New description" \
    --subnet-ids "subnet-42df9c3a" "subnet-48fc21a9"
```

Para Windows:

```
aws elasticache modify-cache-subnet-group ^
    --cache-subnet-group-name mysubnetgroup ^
    --cache-subnet-group-description "New description" ^
```

```
    --subnet-ids "subnet-42df9c3a" "subnet-48fc21a9"
```

Este comando debería producir un resultado similar al siguiente:

```
{
    "CacheSubnetGroup": {
        "VpcId": "vpc-73cd3c17",
        "CacheSubnetGroupDescription": "New description",
        "Subnets": [
            {
                "SubnetIdentifier": "subnet-42dcf93a",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2a"
                }
            },
            {
                "SubnetIdentifier": "subnet-48fc12a9",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2a"
                }
            }
        ],
        "CacheSubnetGroupName": "mysubnetgroup"
    }
}
```

Para obtener más información, consulte el tema AWS CLI de la modify-cache-subnet-group.

Eliminación de un grupo de subredes

Si decide que ya no necesita su grupo de subredes, puede eliminarlo. No puede eliminar un grupo de subredes si hay una caché que lo utiliza actualmente.

Los procedimientos que se describen a continuación muestran cómo eliminar un grupo de subredes.

Eliminación de un grupo de subredes (consola)

Para eliminar un grupo de subredes

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Subnet groups (grupos de subredes).
3. En la lista de grupos de subredes, elija el grupo que desea eliminar y, a continuación, elija Delete(Eliminar).
4. Cuando se le pida que confirme esta operación, escriba el nombre del grupo de subredes en el campo de entrada de texto y elija Eliminar.

Eliminación de un grupo de subredes (AWS CLI)

Mediante la AWS CLI, llame al comando delete-cache-subnet-group con el siguiente parámetro:

- `--cache-subnet-group-name` *`mysubnetgroup`*

Para Linux, macOS o Unix:

```
aws elasticache delete-cache-subnet-group \
    --cache-subnet-group-name mysubnetgroup
```

Para Windows:

```
aws elasticache delete-cache-subnet-group ^
    --cache-subnet-group-name mysubnetgroup
```

Este comando no genera ninguna salida.

Para obtener más información, consulte el tema AWS CLI de la delete-cache-subnet-group.

Identity and Access Management para Amazon ElastiCache

AWS Identity and Access Management (IAM) es un Servicio de AWS que ayuda a los gestionadores a controlar de forma segura el acceso a los recursos de AWS. Los administradores de IAM controlan quién puede estar autenticado (ha iniciado sesión) y autorizado (tiene permisos) para utilizar recursos de ElastiCache. IAM es un servicio de Servicio de AWS que se puedes utilizar sin cargo adicional.

Temas

- Público
- Autenticación con identidades
- Administración de acceso mediante políticas
- Cómo funciona Amazon ElastiCache con IAM
- Ejemplos de políticas basadas en identidades de Amazon ElastiCache
- Solución de problemas de identidad y acceso de Amazon ElastiCache
- Control de acceso
- Información general sobre la administración de los permisos de acceso a los recursos de ElastiCache

Público

La forma en que utiliza AWS Identity and Access Management (IAM) varía en función del rol:

- Usuario del servicio: solicite permisos al administrador si no puede acceder a las características (consulte Solución de problemas de identidad y acceso de Amazon ElastiCache)
- Administrador del servicio: determine el acceso de los usuarios y envíe las solicitudes de permiso (consulte Cómo funciona Amazon ElastiCache con IAM)
- Administrador de IAM: escriba políticas para administrar el acceso (consulte Ejemplos de políticas basadas en identidades de Amazon ElastiCache)

Autenticación con identidades

La autenticación es la manera de iniciar sesión en AWS mediante credenciales de identidad. Debe estar autenticado como Usuario raíz de la cuenta de AWS, un usuario de IAM o asumiendo un rol de IAM.

Puede iniciar sesión como una identidad federada con las credenciales de un origen de identidad, como AWS IAM Identity Center (IAM Identity Center), la autenticación de inicio de sesión único o las credenciales de Google/Facebook. Para obtener más información sobre el inicio de sesión, consulte Cómo iniciar sesión en la Cuenta de AWS en la Guía del usuario de AWS Sign-In.

Para el acceso programático, AWS proporciona un SDK y una CLI para firmar criptográficamente las solicitudes. Para obtener más información, consulte AWS Signature Version 4 para solicitudes de API en la Guía del usuario de IAM.

Usuario raíz de la Cuenta de AWS

Cuando se crea una Cuenta de AWS, se comienza con una identidad de inicio de sesión llamada Cuenta de AWS del usuario raíz que tiene acceso completo a todos los Servicios de AWS y recursos. Recomendamos encarecidamente que no utilice el usuario raíz para las tareas diarias. Para ver las tareas que requieren credenciales de usuario raíz, consulte Tareas que requieren credenciales de usuario raíz en la Guía del usuario de IAM.

Identidad federada

Como práctica recomendada, exija a los usuarios humanos que utilicen la federación con un proveedor de identidades para acceder a Servicios de AWS con credenciales temporales.

Una identidad federada es un usuario del directorio empresarial, proveedor de identidades web o Directory Service que accede a Servicios de AWS mediante credenciales de un origen de identidad. Las identidades federadas asumen roles que proporcionan credenciales temporales.

Para una administración de acceso centralizada, le recomendamos AWS IAM Identity Center. Para obtener más información, consulte ¿Qué es el Centro de identidades de IAM? en la Guía del usuario de AWS IAM Identity Center.

Usuarios y grupos de IAM

Un usuario de IAM es una identidad con permisos específicos para una sola persona o aplicación. Recomendamos el uso de credenciales temporales en lugar de usuarios de IAM con credenciales a

largo plazo. Para obtener más información, consulte Solicitar que los usuarios humanos utilicen la federación con un proveedor de identidades para acceder a AWS mediante credenciales temporales en la Guía del usuario de IAM.

Un grupo de IAM especifica un conjunto de usuarios de IAM y facilita la administración de los permisos para grupos grandes de usuarios. Para obtener más información, consulte Casos de uso para usuarios de IAM en la Guía del usuario de IAM.

Roles de IAM

Un rol de IAM es una identidad con permisos específicos que proporciona credenciales temporales. Puede asumir un rol cambiando de un usuario a un rol de IAM (consola) o llamando a una operación de la API de la AWS CLI o AWS. Para obtener más información, consulte Métodos para asumir un rol en la Guía del usuario de IAM.

Los roles de IAM son útiles para el acceso de usuarios federados, los permisos de usuario de IAM temporales, el acceso entre cuentas, el acceso entre servicios y las aplicaciones que se ejecutan en Amazon EC2. Para obtener más información, consulte Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.

Administración de acceso mediante políticas

Para controlar el acceso en AWS, se crean políticas y se adjuntan a identidades o recursos de AWS. Una política define permisos cuando se asocia con una identidad o recurso. AWS evalúa estas políticas cuando una entidad principal realiza una solicitud. La mayoría de las políticas se almacenan en AWS como documentos JSON. Para obtener más información sobre los documentos de política JSON, consulte Información general de políticas JSON en la Guía del usuario de IAM.

Mediante políticas, los administradores especifican quién tiene acceso a qué definiendo qué entidad principal puede realizar acciones sobre qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM crea políticas de IAM y las agrega a los roles, que luego los usuarios pueden asumir. Las políticas de IAM definen permisos independientemente del método utilizado para realizar la operación.

Políticas basadas en identidades

Las políticas basadas en identidad son documentos JSON de políticas de permisos que asocia a una identidad (usuario, grupo o rol). Estas políticas controlan qué acciones pueden realizar las

identidades, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Las políticas basadas en la identidad pueden ser políticas integradas (incrustadas directamente en una sola identidad) o políticas administradas (políticas independientes asociadas a varias identidades). Para obtener información sobre cómo elegir entre políticas administradas e insertadas, consulte Elegir entre políticas administradas y políticas insertadas en la Guía del usuario de IAM.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de políticas JSON que se asocian a un recurso. Los ejemplos incluyen las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Debe especificar una entidad principal en una política en función de recursos.

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No se puede utilizar políticas de IAM administradas por AWS en una política basada en recursos.

Otros tipos de políticas

AWS admite tipos de políticas adicionales que pueden establecer el máximo de permisos concedidos por los tipos de políticas más comunes:

- Límites de permisos: establecen los permisos máximos que una política basada en la identidad puede conceder a una entidad de IAM. Para obtener más información, consulte Límites de permisos para las entidades de IAM en la Guía del usuario de IAM.
- Políticas de control de servicios (SCP): especifican los permisos máximos para una organización o unidad organizativa en AWS Organizations. Para obtener más información, consulte Políticas de control de servicios en la Guía del usuario de AWS Organizations.
- Políticas de control de recursos (RCP): establecen los permisos máximos disponibles para los recursos de las cuentas. Para obtener más información, consulte Políticas de control de recursos (RCP) en la Guía del usuario de AWS Organizations.
- Políticas de sesión: políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal para un rol o un usuario federado. Para más información, consulte Políticas de sesión en la Guía del usuario de IAM.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para obtener información acerca de cómo AWS decide si permitir o no una solicitud cuando hay varios tipos de políticas implicados, consulte Lógica de evaluación de políticas en la Guía del usuario de IAM.

Cómo funciona Amazon ElastiCache con IAM

Antes de utilizar IAM para administrar el acceso a ElastiCache, conozca qué características de IAM se pueden utilizar con ElastiCache.

Características de IAM que puede utilizar con Amazon ElastiCache

Característica de IAM	Compatibilidad de ElastiCache
Políticas basadas en identidades	Sí
Políticas basadas en recursos	No
Acciones de políticas	Sí
Recursos de políticas	Sí
Claves de condición de política	Sí
ACL	Sí
ABAC (etiquetas en políticas)	Sí
Credenciales temporales	Sí
Permisos de entidades principales	Sí
Roles de servicio	Sí
Roles vinculados al servicio	Sí

Para obtener una perspectiva general sobre cómo funcionan ElastiCache y otros servicios de AWS con la mayoría de las características de IAM, consulte Servicios de AWS que funcionan con IAM en la Guía del usuario de IAM.

Políticas basadas en identidades para ElastiCache

Compatibilidad con las políticas basadas en identidad: sí

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. Para obtener más información sobre los elementos que puede utilizar en una política de JSON, consulte Referencia de los elementos de las políticas de JSON de IAM en la Guía del usuario de IAM.

Ejemplos de políticas basadas en identidades de ElastiCache

Para ver ejemplos de políticas basadas en identidad de ElastiCache, consulte Ejemplos de políticas basadas en identidades de Amazon ElastiCache.

Políticas basadas en recursos en ElastiCache

Admite políticas basadas en recursos: no

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Los ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Las entidades principales pueden incluir cuentas, usuarios, roles, usuarios federados o Servicios de AWS.

Para habilitar el acceso entre cuentas, puedes especificar toda una cuenta o entidades de IAM de otra cuenta como la entidad principal de una política en función de recursos. Para obtener más información, consulte Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.

Acciones de política para ElastiCache

Compatibilidad con las acciones de políticas: sí

Los administradores pueden utilizar las políticas JSON de AWS para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Para ver una lista de las acciones de ElastiCache, consulte Acciones definidas por Amazon ElastiCache, en la Referencia de autorizaciones de servicio.

Las acciones de políticas de ElastiCache utilizan el siguiente prefijo antes de la acción:

```
elasticache
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

```
"Action": [
      "elasticache:action1",
      "elasticache:action2"
         ]
```

Puede utilizar caracteres comodín (*) para especificar varias acciones . Por ejemplo, para especificar todas las acciones que comiencen con la palabra `Describe`, incluya la siguiente acción:

```
"Action": "elasticache:Describe*"
```

Para ver ejemplos de políticas basadas en identidad de ElastiCache, consulte Ejemplos de políticas basadas en identidades de Amazon ElastiCache.

Recursos de política para ElastiCache

Compatibilidad con los recursos de políticas: sí

Los administradores pueden utilizar las políticas JSON de AWS para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Como práctica recomendada, especifique un recurso utilizando el Nombre de recurso de Amazon (ARN). Para las acciones que no admiten permisos por recurso, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

```
"Resource": "*"
```

Para ver una lista de tipos de recursos y ARN de ElastiCache, consulte Tipos de recurso definidos por Amazon ElastiCache, en la Referencia de autorizaciones de servicio. Para obtener información sobre las acciones con las que puede especificar el ARN de cada recurso, consulte Acciones definidas por Amazon ElastiCache.

Para ver ejemplos de políticas basadas en identidad de ElastiCache, consulte Ejemplos de políticas basadas en identidades de Amazon ElastiCache.

Claves de condición de política de ElastiCache

Compatibilidad con claves de condición de políticas específicas del servicio: sí

Los administradores pueden utilizar las políticas JSON de AWS para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` especifica cuándo se ejecutan las instrucciones en función de criterios definidos. Puedes crear expresiones condicionales que utilizan operadores de condición, tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud. Para ver todas las claves de condición globales de AWS, consulte Claves de contexto de condición globales de AWS en la Guía del usuario de IAM.

Para obtener una lista de las claves de condición de ElastiCache, consulte Claves de condición de Amazon ElastiCache en la Referencia de autorizaciones de servicio. Para obtener más información

acerca de las acciones y los recursos con los que puede utilizar una clave de condición, consulte Acciones definidas por Amazon ElastiCache.

Para ver ejemplos de políticas basadas en identidad de ElastiCache, consulte Ejemplos de políticas basadas en identidades de Amazon ElastiCache.

Listas de control de acceso (ACL) en ElastiCache

Compatibilidad con ACL: sí

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Control de acceso basado en atributos (ABAC) con ElastiCache

Admite ABAC (etiquetas en las políticas): sí

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define permisos en función de atributos denominados etiquetas. Puede asociar etiquetas a entidades de IAM y recursos de AWS y, a continuación, diseñar políticas de ABAC para permitir operaciones cuando la etiqueta de la entidad principal coincida con la etiqueta del recurso.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más información sobre ABAC, consulte Definición de permisos con la autorización de ABAC en la Guía del usuario de IAM. Para ver un tutorial con los pasos para configurar ABAC, consulte Uso del control de acceso basado en atributos (ABAC) en la Guía del usuario de IAM.

Uso de credenciales temporales con ElastiCache

Compatibilidad con credenciales temporales: sí

Las credenciales temporales proporcionan acceso a corto plazo a los recursos de AWS y se crean automáticamente cuando se utiliza la federación o se cambia de rol. AWS recomienda que genere

de forma dinámica credenciales temporales en lugar de usar claves de acceso a largo plazo. Para obtener más información, consulte Credenciales de seguridad temporales en IAM y Servicios de AWS que funcionan con IAM en la Guía del usuario de IAM.

Permisos de entidades principales entre servicios de ElastiCache

Admite sesiones de acceso directo (FAS): sí

Las sesiones de acceso directo (FAS) utilizan los permisos de la entidad principal para llamar a un Servicio de AWS, combinados con el Servicio de AWS solicitante para realizar solicitudes a servicios posteriores. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.

Roles de servicio para ElastiCache

Compatibilidad con roles de servicio: sí

Un rol de servicio es un rol de IAM que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.

> **Warning**
>
> Es posible que cambiar los permisos de un rol de servicio interrumpa la funcionalidad de ElastiCache. Edite los roles de servicio solo cuando ElastiCache proporcione orientación para hacerlo.

Roles vinculados a servicios creados para ElastiCache

Admite roles vinculados a servicios: sí

Un rol vinculado a servicios es un tipo de rol de servicio que está vinculado a un Servicio de AWS. El servicio puedes asumir el rol para realizar una acción en su nombre. Los roles vinculados a servicios aparecen en la Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puedes ver, pero no editar, los permisos de los roles vinculados a servicios.

Para más información sobre cómo crear o administrar roles vinculados a servicios, consulta Servicios de AWS que funcionan con IAM. Busque un servicio en la tabla que incluya `Yes` en la columna Rol

vinculado a un servicio. Seleccione el vínculo Sí para ver la documentación acerca del rol vinculado a servicios para ese servicio.

Ejemplos de políticas basadas en identidades de Amazon ElastiCache

De forma predeterminada, los usuarios y roles no tienen permiso para crear ni modificar los recursos de ElastiCache. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan.

Para obtener información acerca de cómo crear una política basada en identidades de IAM mediante el uso de estos documentos de políticas JSON de ejemplo, consulte Creación de políticas de IAM (consola) en la Guía del usuario de IAM.

Para obtener más información sobre las acciones y los tipos de recursos definidos por ElastiCache, incluido el formato de los ARN para cada tipo de recurso, consulte Acciones, recursos y claves de condición para Amazon ElastiCache, en la Referencia de autorizaciones de servicio.

Temas

- Prácticas recomendadas sobre las políticas
- Uso de la consola de ElastiCache
- Cómo permitir a los usuarios consultar sus propios permisos

Prácticas recomendadas sobre las políticas

Las políticas basadas en identidades determinan si alguien puede crear, acceder o eliminar los recursos de ElastiCache de la cuenta. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comienza con las políticas administradas por AWSy continúa con los permisos de privilegio mínimo: a fin de comenzar a conceder permisos a los usuarios y las cargas de tarea, utiliza las políticas administradas por AWS, que conceden permisos para muchos casos de uso comunes. Están disponibles en su Cuenta de AWS. Se recomienda definir políticas gestionadas por el cliente de AWS específicas para sus casos de uso a fin de reducir aún más los permisos. Con el fin de obtener más información, consulte las políticas administradas por AWS o las políticas administradas por AWS para funciones de tarea en la Guía de usuario de IAM.
- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se

pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulte Políticas y permisos en IAM en la Guía del usuario de IAM.

- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puedes escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puedes usar condiciones para conceder acceso a acciones de servicios si se emplean a través de un Servicio de AWS determinado como, por ejemplo, CloudFormation. Para obtener más información, consulte Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.
- Utiliza el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte Validación de políticas con el Analizador de acceso de IAM en la Guía del usuario de IAM.
- Solicite la autenticación multifactor (MFA): si se encuentra en una situación en la que necesite usuarios raíz o de IAM en su Cuenta de AWS, active la MFA para obtener una mayor seguridad. Para exigir la MFA cuando se invoquen las operaciones de la API, añada condiciones de MFA a sus políticas. Para más información, consulte Acceso seguro a la API con MFA en la Guía del usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.

Uso de la consola de ElastiCache

Para acceder a la consola de Amazon ElastiCache, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle mostrar y consultar los detalles acerca de los recursos de ElastiCache en la Cuenta de AWS. Si crea una política basada en identidades que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para las entidades (usuarios o roles) que tengan esa política.

No es necesario que conceda permisos mínimos para la consola a los usuarios que solo realizan llamadas a la AWS CLI o a la API de AWS. En su lugar, permite el acceso únicamente a las acciones que coincidan con la operación de API que intentan realizar.

Para asegurarse de que los usuarios y los roles puedan seguir utilizando la consola de ElastiCache, asocie también la política administrada `ConsoleAccess` o `ReadOnly` de AWS de ElastiCache a las entidades. Para obtener más información, consulte Adición de permisos a un usuario en la Guía del usuario de IAM:

Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas administradas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para llevar a cabo esta acción en la consola o mediante programación con la AWS CLI o la API de AWS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
```

```
    ]
}
```

Solución de problemas de identidad y acceso de Amazon ElastiCache

Utilice la siguiente información para diagnosticar y solucionar los problemas comunes que puedan surgir cuando trabaje con ElastiCache e IAM.

Temas

- No tengo autorización para realizar una acción en ElastiCache
- No tengo autorización para realizar la operación iam:PassRole
- Deseo permitir que personas ajenas a mi cuenta de AWS puedan acceder a mis recursos de ElastiCache

No tengo autorización para realizar una acción en ElastiCache

Si la Consola de administración de AWS le indica que no está autorizado para llevar a cabo una acción, debe ponerse en contacto con su administrador para recibir ayuda. Su administrador es la persona que le facilitó su nombre de usuario y contraseña.

En el siguiente ejemplo, el error se produce cuando el usuario `mateojackson` intenta utilizar la consola para consultar los detalles acerca de un recurso ficticio *`my-example-widget`*, pero no tiene los permisos ficticios `elasticache:`*`GetWidget`*.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 elasticache:GetWidget on resource: my-example-widget
```

En este caso, Mateo pide a su administrador que actualice sus políticas de forma que pueda obtener acceso al recurso *`my-example-widget`* mediante la acción `elasticache:`*`GetWidget`*.

No tengo autorización para realizar la operación iam:PassRole

Si recibe un error que indica que no tiene autorización para realizar la acción `iam:PassRole`, las políticas se deben actualizar a fin de permitirle pasar un rol a ElastiCache.

Algunos Servicios de AWS le permiten transferir un rol existente a dicho servicio en lugar de crear un nuevo rol de servicio o uno vinculado al servicio. Para ello, debe tener permisos para transferir el rol al servicio.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en ElastiCache. Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con su gestionador de AWS. El gestionador es la persona que le proporcionó las credenciales de inicio de sesión.

Deseo permitir que personas ajenas a mi cuenta de AWS puedan acceder a mis recursos de ElastiCache

Puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Puede especificar una persona de confianza para que asuma el rol. En el caso de los servicios que admitan las políticas basadas en recursos o las listas de control de acceso (ACL), puede utilizar dichas políticas para conceder a las personas acceso a sus recursos.

Para obtener más información, consulte lo siguiente:

- Para obtener información acerca de si ElastiCache admite estas características, consulte Cómo funciona Amazon ElastiCache con IAM.
- Para obtener información acerca de cómo proporcionar acceso a los recursos de las Cuentas de AWS de su propiedad, consulte Acceso para un usuario de IAM en otra Cuenta de AWS propia en la Guía del usuario de IAM.
- Para obtener información acerca de cómo proporcionar acceso a tus recursos a Cuentas de AWS de terceros, consulta Proporcionar acceso a Cuentas de AWS que son propiedad de terceros en la Guía del usuario de IAM.

- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulta Proporcionar acceso a usuarios autenticados externamente (identidad federada) en la Guía del usuario de IAM.
- Para conocer sobre la diferencia entre las políticas basadas en roles y en recursos para el acceso entre cuentas, consulte Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.

Control de acceso

Puede tener credenciales válidas para autenticar las solicitudes, pero a menos que tenga permisos no puede crear o acceder a los recursos de ElastiCache. Por ejemplo, debe tener permisos para crear un clúster de ElastiCache.

En las siguientes secciones, se describe cómo administrar los permisos de ElastiCache. Recomendamos que lea primero la información general.

- Información general sobre la administración de los permisos de acceso a los recursos de ElastiCache
- Uso de políticas basadas en identidad (políticas de IAM) para Amazon ElastiCache

Información general sobre la administración de los permisos de acceso a los recursos de ElastiCache

Cada recurso de AWS es propiedad de una cuenta de AWS y los permisos para crear o tener acceso a un recurso se rigen por las políticas de permisos. Un administrador de cuentas puede asociar políticas de permisos a identidades de IAM (es decir, usuarios, grupos y funciones). Además, Amazon ElastiCache también permite adjuntar políticas de permisos a los recursos.

> **Note**
>
> Un administrador de cuentas (o usuario administrador) es un usuario que tiene privilegios de administrador. Para obtener más información, consulte Prácticas recomendadas de IAM en la Guía del usuario de IAM.

Para dar acceso, agregue permisos a los usuarios, grupos o roles:

- Usuarios y grupos en AWS IAM Identity Center:

  Cree un conjunto de permisos. Siga las instrucciones de Creación de un conjunto de permisos en la Guía del usuario de AWS IAM Identity Center.
- Usuarios gestionados en IAM a través de un proveedor de identidades:

  Cree un rol para la federación de identidades. Siga las instrucciones descritas en Creación de un rol para un proveedor de identidad de terceros (federación) en la Guía del usuario de IAM.
- Usuarios de IAM:
  - Cree un rol que el usuario pueda aceptar. Siga las instrucciones descritas en Creación de un rol para un usuario de IAM en la Guía del usuario de IAM.
  - (No recomendado) Adjunte una política directamente a un usuario o añada un usuario a un grupo de usuarios. Siga las instrucciones descritas en Adición de permisos a un usuario (consola) de la Guía del usuario de IAM.

Temas

- Operaciones y recursos de Amazon ElastiCache
- Titularidad de los recursos
- Administración del acceso a los recursos

- Políticas administradas por AWS para Amazon ElastiCache
- Uso de políticas basadas en identidad (políticas de IAM) para Amazon ElastiCache
- Permisos de nivel de recursos
- Uso de claves de condición
- Uso de roles vinculados a servicios para Amazon ElastiCache
- Permisos de la API de ElastiCache: referencia de condiciones, acciones y recursos

Operaciones y recursos de Amazon ElastiCache

Para ver una lista de tipos de recursos y ARN de ElastiCache, consulte Tipos de recurso definidos por Amazon ElastiCache, en la Referencia de autorizaciones de servicio. Para obtener información sobre las acciones con las que puede especificar el ARN de cada recurso, consulte Acciones definidas por Amazon ElastiCache.

Titularidad de los recursos

El propietario de los recursos es la cuenta de AWS que crea el recurso. Es decir, el propietario del recurso es la cuenta de AWS de la entidad principal que autentica la solicitud que creó el recurso. Una entidad principal puede ser la cuenta raíz, un usuario de IAM o un rol de IAM. Los siguientes ejemplos ilustran cómo funciona:

- Supongamos que utiliza las credenciales de la cuenta raíz de la cuenta de AWS para crear un clúster de caché. En este caso, la cuenta de AWS es la propietaria del recurso. En ElastiCache, el recurso es el clúster.
- Supongamos que crea un usuario de IAM en la cuenta de AWS y concede permisos a fin de crear un clúster para dicho usuario. En este caso, el usuario puede crear un clúster. Sin embargo, la cuenta de AWS, a la que pertenece el usuario, es la propietaria del recurso de clúster.
- Supongamos que crea un rol de IAM en la cuenta de AWS con permisos para crear un clúster. En este caso, cualquiera que pueda asumir el rol puede crear un clúster. La cuenta de AWS, a la que pertenece el rol, es la propietaria del recurso de clúster de caché.

Administración del acceso a los recursos

Una política de permisos describe quién tiene acceso a qué. En la siguiente sección se explican las opciones disponibles para crear políticas de permisos.

> **Note**
>
> En esta sección, también se explica el uso de IAM en el contexto de Amazon ElastiCache. No se proporciona información detallada sobre el servicio de IAM. Para ver la documentación completa de IAM, consulte ¿Qué es IAM? en la Guía del usuario de IAM. Para obtener más información acerca de la sintaxis y las descripciones de las políticas del IAM, consulte Referencia de políticas de IAM de AWS en la Guía del usuario de IAM.

Las políticas que se asocian a una identidad de IAM se denominan políticas basadas en identidades (o políticas de IAM). Las políticas que se adjuntan a un recurso se denominan políticas basadas en recursos.

Temas

- Políticas basadas en identidades (políticas de IAM)
- Especificación de elementos de política: acciones, efectos, recursos y entidades principales
- Especificación de las condiciones de una política

Políticas basadas en identidades (políticas de IAM)

Puede asociar políticas a identidades de IAM. Por ejemplo, puede hacer lo siguiente:

- Asociar una política de permisos a un usuario o grupo de la cuenta: un administrador de la cuenta puede utilizar una política de permisos asociada a un usuario determinado para concederle permisos. En este caso, los permisos son para que ese usuario cree un recurso de ElastiCache, como un clúster, un grupo de parámetros o un grupo de seguridad.
- Asociar una política de permisos a un rol (conceder permisos entre cuentas): puede asociar una política de permisos basada en identidad a un rol de &IAM; para conceder permisos entre cuentas. Por ejemplo, el administrador de la Cuenta A puede crear un rol para conceder permisos entre cuentas a otra cuenta de AWS (por ejemplo, a la Cuenta B) o a un servicio de AWS, tal y como se indica a continuación:
  1. El administrador de la CuentaA crea un rol de IAM y asocia una política de permisos a dicho rol, que concede permisos sobre los recursos de la CuentaA.
  2. El administrador de la CuentaA asocia una política de confianza al rol que identifica la Cuenta B como la entidad principal que puede asumir el rol.

3. El administrador de la Cuenta B puede delegar permisos para asumir el rol a cualquier usuario de la Cuenta B. Hacer esto permite a los usuarios de la Cuenta B crear o acceder a los recursos de la Cuenta A. En algunos casos, es posible que desee conceder a un servicio de AWS permisos para asumir el rol. Para respaldar este enfoque, la entidad principal de la política de confianza también puede ser la entidad principal de un servicio de AWS.

Para obtener más información sobre el uso de IAM para delegar permisos, consulte Administración de accesos en la Guía del usuario de IAM.

A continuación se ofrece un ejemplo de una política que permite a un usuario realizar la acción `DescribeCacheClusters` para la cuenta de AWS. ElastiCache también permite identificar recursos específicos mediante los ARN de recurso para realizar acciones de la API. Este enfoque también se conoce como "permisos a nivel de recursos".

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "DescribeCacheClusters",
            "Effect": "Allow",
            "Action": [
                "elasticache:DescribeCacheClusters"
            ],
            "Resource": "arn:aws:iam::*:role/*"
        }
    ]
}
```

Para obtener más información sobre el uso de políticas basadas en identidades con ElastiCache, consulte Uso de políticas basadas en identidad (políticas de IAM) para Amazon ElastiCache. Para obtener más información sobre usuarios, grupos, roles y permisos, consulte Identidades (usuarios, grupos y roles) en la Guía del usuario de IAM.

Especificación de elementos de política: acciones, efectos, recursos y entidades principales

En cada recurso de Amazon ElastiCache (consulte Operaciones y recursos de Amazon ElastiCache), el servicio define un conjunto de operaciones de la API (consulte Acciones). A fin de conceder permisos para estas operaciones de API, ElastiCache define un conjunto de acciones que puede especificar en una política. Por ejemplo, para el recurso del clúster de ElastiCache, se definen las siguientes acciones: `CreateCacheCluster`, `DeleteCacheCluster` y `DescribeCacheCluster`. Para realizar una operación API pueden ser necesarios permisos para más de una acción.

A continuación se indican los elementos más básicos de la política:

- Recurso: en una política, se usa un nombre de recurso de Amazon (ARN) para identificar el recurso al que se aplica la política. Para obtener más información, consulte Operaciones y recursos de Amazon ElastiCache.
- Acción: utilice palabras clave de acción para identificar las operaciones del recurso que desea permitir o denegar. Por ejemplo, en función del elemento `Effect` especificado, el permiso `elasticache:CreateCacheCluster` permite o deniega al usuario los permisos para realizar la operación `CreateCacheCluster` de Amazon ElastiCache.
- Efecto: especifique el efecto que se producirá cuando el usuario solicite la acción específica; puede ser permitir o denegar. Si no concede acceso de forma explícita (permitir) a un recurso, el acceso se deniega implícitamente. También puede denegar explícitamente el acceso a un recurso. Por ejemplo, esto puede servir para asegurarse de que un usuario no pueda tener acceso al recurso, aunque otra política le conceda acceso.
- Entidad principal: en las políticas basadas en identidades (políticas de IAM), el usuario al que se asocia esta política es la entidad principal implícita. Para las políticas basadas en recursos, debe especificar el usuario, la cuenta, el servicio u otra entidad que desee que reciba permisos (se aplica solo a las políticas basadas en recursos).

Para obtener más información sobre la sintaxis y descripciones de las políticas de IAM, consulte Referencia de la política de IAM de AWS en la Guía del usuario de IAM.

Para ver una tabla con todas las acciones de API de Amazon ElastiCache, consulte Permisos de la API de ElastiCache: referencia de condiciones, acciones y recursos.

Especificación de las condiciones de una política

Al conceder permisos, puede utilizar el lenguaje de la política de IAM para especificar las condiciones en la que se debe aplicar una política. Por ejemplo, es posible que desee que solo se aplique una

política después de una fecha específica. Para obtener más información sobre cómo especificar condiciones en un lenguaje de política, consulte Condition en la Guía del usuario de IAM.

Para expresar condiciones, se usan claves de condición predefinidas. Para utilizar las claves de condición específicas de ElastiCache, consulte Uso de claves de condición. Existen claves de condición que se aplican a todo AWS que puede utilizar cuando corresponda. Para ver una lista completa de las claves que se aplican a todo AWS, consulte Claves disponibles para las condiciones en la Guía del usuario de IAM.

Políticas administradas por AWS para Amazon ElastiCache

Una política administrada de AWS es una política independiente que AWS crea y administra. Las políticas administradas de AWS se diseñan para ofrecer permisos para muchos casos de uso comunes, por lo que puede empezar a asignar permisos a los usuarios, grupos y roles.

Considere que es posible que las políticas administradas de AWS no concedan permisos de privilegio mínimo para los casos de uso concretos, ya que están disponibles para que las utilicen todos los clientes de AWS. Se recomienda definir políticas administradas por el cliente específicas para sus casos de uso a fin de reducir aún más los permisos.

No puede cambiar los permisos definidos en las políticas administradas de AWS. Si AWS actualiza los permisos definidos en una política administrada de AWS, la actualización afecta a todas las identidades de entidades principales (usuarios, grupos y roles) a las que está adjunta la política. Lo más probable es que AWS actualice una política administrada de AWS cuando se lance un nuevo Servicio de AWS o las operaciones de la API nuevas estén disponibles para los servicios existentes.

Para obtener más información, consulte Políticas administradas de AWS en la Guía del usuario de IAM.

Política administrada por AWS: ElastiCacheServiceRolePolicy

No puede adjuntar ElastiCacheServiceRolePolicy a las entidades de IAM. Esta política está adjunta a un rol vinculado a servicios que permite a ElastiCache realizar acciones en su nombre.

Esta política permite que ElastiCache administre recursos de AWS en su nombre, según sea necesario para administrar la caché:

- `ec2`: administre los recursos de red de EC2 para asociarlos a los nodos de caché, incluidos los puntos de conexión de VPC (para cachés sin servidor), las interfaces de red elásticas (ENI) (para clústeres basados en nodos) y los grupos de seguridad.
- `cloudwatch`: emita datos sobre métricas del servicio a CloudWatch.
- `outposts`: permita la creación de nodos de caché en AWS Outposts.

Puede encontrar la política ElastiCacheServiceRolePolicy en la consola IAM, y puede encontrar la política ElastiCacheServiceRolePolicy en la Guía de referencia de políticas administradas de AWS.

Política administrada por AWS: AmazonElastiCacheFullAccess

Puede adjuntar la política `AmazonElastiCacheFullAccess` a las identidades de IAM.

Esta política les da a las entidades principales el acceso total a ElastiCache mediante la consola de administración de AWS:

- `elasticache`: acceso completo a todas las API.
- `iam`: cree el rol vinculado al servicio necesario para el funcionamiento del servicio.
- `ec2`: describa los recursos de EC2 dependientes necesarios para la creación de caché (VPC, subred, grupo de seguridad) y permita la creación de puntos de conexión de VPC (para cachés sin servidor).
- `kms`: permita el uso de CMK administradas por el cliente para el cifrado en reposo.
- `cloudwatch`: permita el acceso a métricas a fin de mostrar las métricas de ElastiCache en la consola.
- `application-autoscaling`: permita el acceso para describir las políticas de escalado automático de las cachés.
- `logs`: se utiliza para rellenar los flujos de registros para la funcionalidad de entrega de registros en la consola.
- `firehose`: se utiliza para rellenar los flujos para la funcionalidad de entrega de registros en la consola.
- `s3`: se utiliza a fin de rellenar los buckets S3 para la funcionalidad de restauración de instantáneas en la consola.
- `outposts`: se utiliza a fin de rellenar AWS Outposts para la creación de cachés en la consola.
- `sns`: se utiliza para rellenar los temas de SNS para la funcionalidad de notificación en la consola.

Puede encontrar la política AmazonElastiCacheFullAccess en la consola de IAM y puede encontrar la política AmazonElastiCacheFullAccess en la Guía de referencia de las políticas administradas por AWS.

Política administrada por AWS: AmazonElastiCacheReadOnlyAccess

Puede adjuntar la política `AmazonElastiCacheReadOnlyAccess` a las identidades de IAM.

Esta política les da a las entidades principales el acceso de solo lectura a ElastiCache mediante la consola de administración de AWS:

- `elasticache`: acceda a las API `Describe` de solo lectura.

Puede encontrar la política AmazonElastiCacheReadOnlyAccess en la consola de IAM y puede encontrar la política AmazonElastiCacheReadOnlyAccess en la Guía de referencia de las políticas administradas por AWS.

Actualizaciones de ElastiCache a las políticas administradas de AWS

Es posible consultar los detalles sobre las actualizaciones de las políticas administradas de AWS para ElastiCache debido a que este servicio comenzó a realizar un seguimiento de estos cambios. Para obtener alertas automáticas sobre los cambios realizados en esta página, suscríbase a la fuente RSS en la página del historial de documentos de ElastiCache.

Cambio	Descripción	Fecha
AmazonElastiCacheFullAccess: actualización de una política existente.	ElastiCache ha añadido nuevos permisos para permitir el escalado vertical de MemCached para la acción `elasticache:ModifyCacheCluster` .	27 de marzo de 2025
AmazonElastiCacheFullAccess: actualización de una política existente.	ElastiCache agregó nuevos permisos para permitir la administración de cachés sin servidor y para permitir el uso de todas las características del servicio a través de la consola.	27 de noviembre de 2023
ElastiCacheServiceRolePolicy: actualización de una política existente.	ElastiCache agregó nuevos permisos para permitir la administración de los puntos de conexión de VPC para	27 de noviembre de 2023

Cambio	Descripción	Fecha
	los recursos de caché sin servidor.	
ElastiCache comenzó a hacer un seguimiento de los cambios	ElastiCache comenzó a realizar el seguimiento de los cambios de las políticas administradas de AWS.	7 de febrero de 2020

Uso de políticas basadas en identidad (políticas de IAM) para Amazon ElastiCache

Este tema contiene ejemplos de políticas basadas en identidades, donde los administradores de cuentas pueden asociar políticas de permisos a identidades de IAM (es decir, a usuarios, grupos y funciones).

> **Important**
>
> Recomendamos que primero lea los temas en los que se explican los conceptos básicos y las opciones para administrar el acceso a los recursos de Amazon ElastiCache. Para obtener más información, consulte Información general sobre la administración de los permisos de acceso a los recursos de ElastiCache.

En las secciones de este tema se explica lo siguiente:

- Políticas administradas por AWS para Amazon ElastiCache
- Ejemplos de políticas administradas por los clientes

A continuación, se incluye un ejemplo de una política de permisos cuando se utiliza Redis OSS.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
```

```
            "Sid": "AllowClusterPermissions",
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateServerlessCache",
                "elasticache:CreateCacheCluster",
                "elasticache:DescribeServerlessCaches",
                "elasticache:DescribeReplicationGroups",
                "elasticache:DescribeCacheClusters",
                "elasticache:ModifyServerlessCache",
                "elasticache:ModifyReplicationGroup",
                "elasticache:ModifyCacheCluster"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUserToPassRole",
            "Effect": "Allow",
            "Action": [ "iam:PassRole" ],
            "Resource": "arn:aws:iam::123456789012:role/EC2-roles-for-cluster"
        }
    ]
}
```

A continuación, se muestra un ejemplo de una política de permisos cuando se utiliza Memcached.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Sid": "AllowClusterPermissions",
        "Effect": "Allow",
        "Action": [
            "elasticache:CreateServerlessCache",
            "elasticache:CreateCacheCluster",
            "elasticache:DescribeServerlessCaches",
            "elasticache:DescribeCacheClusters",
            "elasticache:ModifyServerlessCache",
            "elasticache:ModifyCacheCluster"
        ],
        "Resource": "*"
    },
```

```
    {
        "Sid": "AllowUserToPassRole",
        "Effect": "Allow",
        "Action": [ "iam:PassRole" ],
        "Resource": "arn:aws:iam::123456789012:role/EC2-roles-for-cluster"
    }
    ]
}
```

La política tiene dos instrucciones:

- La primera declaración concede permisos para las acciones de Amazon ElastiCache (`elasticache:Create*`, `elasticache:Describe*`, `elasticache:Modify*`)
- La segunda declaración concede permisos para la acción de IAM (`iam:PassRole`) en el nombre de rol de IAM especificado al final del valor `Resource`.

La política no especifica el elemento `Principal`, ya que en una política basada en la identidad no se especifica el elemento principal que obtiene el permiso. Al asociar una política a un usuario, el usuario es la entidad principal implícita. Cuando asocia una política de permisos a un rol de IAM, el elemento principal identificado en la política de confianza de rol obtiene los permisos.

Para ver una tabla con todas las acciones de la API de Amazon ElastiCache y los recursos a los que se aplican, consulte Permisos de la API de ElastiCache: referencia de condiciones, acciones y recursos.

Ejemplos de políticas administradas por los clientes

Si no está utilizando una política predeterminada y elige utilizar una política administrada de forma personalizada, asegúrese de una de las dos cosas. Debería tener permisos para llamar a `iam:createServiceLinkedRole` (para obtener más información, consulte Ejemplo 4: Permitir que un usuario llame a la API CreateServiceLinkedRole de IAM). También debe haber creado un rol vinculado al servicio de ElastiCache.

Combinadas con los permisos mínimos necesarios para utilizar la consola de Amazon ElastiCache, las políticas de ejemplo de esta sección conceden permisos adicionales. Los ejemplos también son relevantes para los SDK de AWS y la AWS CLI.

Para obtener instrucciones sobre la configuración de grupos y usuarios de IAM, consulte Creación del primer grupo y usuario administrador de IAM en la Guía del usuario de IAM.

> **Important**
>
> Pruebe siempre sus políticas de IAM antes de utilizarlas en entornos de producción. Algunas acciones de ElastiCache que parecen sencillas pueden requerir otras acciones de apoyo cuando se utiliza la consola de ElastiCache. Por ejemplo, `elasticache:CreateCacheCluster` concede permisos para crear clústeres de ElastiCache. Sin embargo, para realizar esta operación, la consola de ElastiCache utiliza varias acciones `Describe` y `List` a fin de completar las listas de la consola.

Ejemplos

- Ejemplo 1: permitir al usuario acceso de solo lectura a los recursos de ElastiCache
- Ejemplo 2: conceder a un usuario permiso para realizar tareas comunes de administrador del sistema de ElastiCache
- Ejemplo 3: conceder a un usuario permiso para tener acceso a todas las acciones de API de ElastiCache
- Ejemplo 4: Permitir que un usuario llame a la API CreateServiceLinkedRole de IAM
- Ejemplo 5: permitir que un usuario se conecte a una caché sin servidor mediante la autenticación de IAM

Ejemplo 1: permitir al usuario acceso de solo lectura a los recursos de ElastiCache

La siguiente política concede permisos para utilizar acciones de ElastiCache que permiten a un usuario mostrar recursos. Normalmente, este tipo de política de permisos se adjunta a un grupo de administradores.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[{
      "Sid": "ECReadOnly",
      "Effect":"Allow",
      "Action": [
          "elasticache:Describe*",
          "elasticache:List*"],
      "Resource":"*"
```

```
      }
   ]
}
```

Ejemplo 2: conceder a un usuario permiso para realizar tareas comunes de administrador del sistema de ElastiCache

Las tareas comunes del administrador del sistema incluyen la modificación de los recursos. También es posible que el administrador del sistema necesite obtener información acerca de los eventos de ElastiCache. La siguiente política concede a un usuario permisos a fin de realizar acciones de ElastiCache para estas tareas comunes de administrador del sistema. Normalmente, este tipo de política de permisos se adjunta al grupo de administradores del sistema.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[{
      "Sid": "ECAllowMutations",
      "Effect":"Allow",
      "Action":[
          "elasticache:Modify*",
          "elasticache:Describe*",
          "elasticache:ResetCacheParameterGroup"
      ],
      "Resource":"*"
      }
   ]
}
```

Ejemplo 3: conceder a un usuario permiso para tener acceso a todas las acciones de API de ElastiCache

La siguiente política permite a un usuario acceder a todas las acciones de ElastiCache. Recomendamos que conceda este tipo de política de permisos solo a un usuario administrador.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[{
      "Sid": "ECAllowAll",
      "Effect":"Allow",
      "Action":[
          "elasticache:*"
      ],
      "Resource":"*"
      }
   ]
}
```

Ejemplo 4: Permitir que un usuario llame a la API CreateServiceLinkedRole de IAM

La siguiente política permite al usuario llamar a la API `CreateServiceLinkedRole` de IAM. Recomendamos que conceda este tipo de política de permisos al usuario que invoca las operaciones de ElastiCache mutantes.

JSON

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid":"CreateSLRAllows",
      "Effect":"Allow",
      "Action":[
        "iam:CreateServiceLinkedRole"
      ],
      "Resource":"*",
      "Condition":{
        "StringLike":{
        "iam:AWSServiceName":"elasticache.amazonaws.com"
        }
      }
    }
  ]
```

```
}
```

Ejemplo 5: permitir que un usuario se conecte a una caché sin servidor mediante la autenticación de IAM

La siguiente política permite que cualquier usuario se conecte a cualquier caché sin servidor mediante la autenticación de IAM entre el 1 de abril de 2023 y el 30 de junio de 2023.

JSON

```
{
  "Version":"2012-10-17",
  "Statement" :
  [
    {
      "Effect" : "Allow",
      "Action" : ["elasticache:Connect"],
      "Resource" : [
        "arn:aws:elasticache:us-east-1:123456789012:serverlesscache:*"
      ],
      "Condition": {
        "DateGreaterThan": {"aws:CurrentTime": "2023-04-01T00:00:00Z"},
        "DateLessThan": {"aws:CurrentTime": "2023-06-30T23:59:59Z"}
      }
    },
    {
      "Effect" : "Allow",
      "Action" : ["elasticache:Connect"],
      "Resource" : [
        "arn:aws:elasticache:us-east-1:123456789012:user:*"
      ]
    }
  ]
}
```

Permisos de nivel de recursos

Puede restringir el alcance de los permisos de un usuario mediante la especificación de recursos en una política de IAM. Muchas acciones de la API de ElastiCache admiten un tipo de recurso que varía en función del comportamiento de la acción. Cada instrucción de una política de IAM

concede permiso para realizar una acción en un recurso. Cuando la acción no actúa sobre un recurso nombrado, o cuando se concede permiso para realizar la acción sobre todos los recursos, el valor del recurso en la política es un comodín (*). Para muchas acciones de API, puede restringir los recursos que un usuario puede modificar si especifica el nombre de recurso de Amazon (ARN) de un recurso o un patrón de ARN que coincida con varios recursos. Para restringir los permisos por recurso, especifique el recurso por ARN.

Para ver una lista de tipos de recursos y ARN de ElastiCache, consulte Tipos de recurso definidos por Amazon ElastiCache, en la Referencia de autorizaciones de servicio. Para obtener información sobre las acciones con las que puede especificar el ARN de cada recurso, consulte Acciones definidas por Amazon ElastiCache.

Ejemplos

- Ejemplo 1: permitir a un usuario acceso completo a tipos de recursos específicos de ElastiCache
- Ejemplo 2: denegarle a un usuario el acceso a una memoria caché sin servidor.

Ejemplo 1: permitir a un usuario acceso completo a tipos de recursos específicos de ElastiCache

La siguiente política permite de forma explícita todos los recursos del tipo caché sin servidor.

```
{
        "Sid": "Example1",
        "Effect": "Allow",
        "Action": "elasticache:*",
        "Resource": [
             "arn:aws:elasticache:us-east-1:account-id:serverlesscache:*"
        ]
}
```

Ejemplo 2: denegarle a un usuario el acceso a una memoria caché sin servidor.

En el siguiente ejemplo, se deniega de forma explícita el acceso a una determinada caché sin servidor.

```
{
        "Sid": "Example2",
        "Effect": "Deny",
        "Action": "elasticache:*",
        "Resource": [
```

```
            "arn:aws:elasticache:us-east-1:account-id:serverlesscache:name"
        ]
}
```

Uso de claves de condición

Puede especificar condiciones que determinan cómo se aplica una política de IAM. En ElastiCache, puede utilizar el elemento `Condition` de una política JSON para comparar las claves en el contexto de la solicitud con los valores de las claves que especifique en la política. Para obtener más información, consulte Elementos de la política de JSON de IAM: Condición.

Para obtener una lista de las claves de condición de ElastiCache, consulte Claves de condición de Amazon ElastiCache en la Referencia de autorizaciones de servicio.

Para obtener una lista de todas las claves de condición globales, consulte Claves de contexto de condición globales de AWS.

Uso de ElastiCache con claves de condición globales de AWS

Cuando utilice claves de condición globales de AWS que requieran la Entidad principal de ElastiCache, utilice una condición `OR` con ambas entidades principales: `elasticache.amazonaws.com` y `ec.amazonaws.com`.

> **Note**
>
> Si no añade ambas entidades principales a ElastiCache, la acción prevista de permitir o denegar no se aplicará correctamente a ninguno de los recursos incluidos en su política.

Ejemplo de política con una clave de condición global de `aws:CalledVia`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:*",
      "Resource": "*",
```

```
      "Condition": {
        "ForAnyValue:StringLike": {
          "aws:CalledVia": [
            "ec.amazonaws.com",
            "elasticache.amazonaws.com"
          ]
        }
      }
    }
  ]
}
```

Especificación de condiciones: uso de claves de condición

Para implementar el control detallado, hay que escribir una política de permisos de IAM que especifique las condiciones a fin de controlar un conjunto de parámetros individuales en determinadas solicitudes. A continuación, se aplica la política de IAM a los usuarios, los grupos o los roles creados con la consola de IAM.

Para aplicar una condición, agregue la información de condición a la declaración de la política de IAM. En el siguiente ejemplo, se especifica la condición de que cualquier clúster basado en nodos que se cree será del tipo de nodo `cache.r5.large`.

> **Note**
>
> - Para construir elementos `Condition` utilizando claves de condición del tipo `String`, utilice los operadores de condición que no distinguen mayúsculas de minúsculas `StringEqualsIgnoreCase` o `StringNotEqualsIgnoreCase` para comparar una clave con un valor de cadena.
> - ElastiCache procesa los argumentos de entrada para `CacheNodeType` y `CacheParameterGroupName` sin distinguir entre mayúsculas y minúsculas. Por este motivo, los operadores de condición de cadena `StringEqualsIgnoreCase` y `StringNotEqualsIgnoreCase` deben usarse en las políticas de permisos que hacen referencia a ellos.

A continuación, se incluye un ejemplo de esta política de permisos cuando se utiliza Valkey o Redis OSS.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*",
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "StringEquals": {
                    "elasticache:CacheNodeType": [
                        "cache.r5.large"
                    ]
                }
            }
        }
    ]
}
```

A continuación, se muestra un ejemplo de esta política de permisos cuando se utiliza Memcached.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*"
            ],
            "Condition": {
                "StringEquals": {
                    "elasticache:CacheNodeType": [
                        "cache.r5.large"
                    ]
                }
            }
        }
    ]
}
```

Para obtener más información, consulte Etiquetado de los recursos de ElastiCache.

Para obtener más información sobre el uso de operadores de condición de política, consulte Permisos de la API de ElastiCache: referencia de condiciones, acciones y recursos.

Ejemplos de políticas: uso de condiciones para el control de parámetros de precisión

En esta sección se muestran las políticas de ejemplo para implementar un control de parámetros de precisión en los parámetros de ElastiCache enumerados con anterioridad.

1. elastiCache:MaximumDataStorage: especifique el almacenamiento máximo de datos de una caché sin servidor. Mediante las condiciones proporcionadas, el cliente no puede crear cachés que puedan almacenar más de una cantidad específica de datos.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowDependentResources",
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateServerlessCache"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:serverlesscachesnapshot:*",
                "arn:aws:elasticache:*:*:snapshot:*",
                "arn:aws:elasticache:*:*:usergroup:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateServerlessCache"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:serverlesscache:*"
            ],
            "Condition": {
                "NumericLessThanEquals": {
                    "elasticache:MaximumDataStorage": "30"
                },
                "StringEquals": {
                    "elasticache:DataStorageUnit": "GB"
                }
            }
        }
```

```
    ]
}
```

2. elasticache:MaximumECPUPerSecond: especifique el valor máximo de ECPU por segundo de una caché sin servidor. Mediante las condiciones proporcionadas, el cliente no puede crear cachés que puedan ejecutar más de un número específico de ECPU por segundo.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowDependentResources",
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateServerlessCache"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:serverlesscachesnapshot:*",
                "arn:aws:elasticache:*:*:snapshot:*",
                "arn:aws:elasticache:*:*:usergroup:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateServerlessCache"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:serverlesscache:*"
            ],
            "Condition": {
                "NumericLessThanEquals": {
                    "elasticache:MaximumECPUPerSecond": "100000"
                }
            }
        }
    ]
}
```

3. elasticache:CacheNodeType: especifique qué NodeType puede crear un usuario. Mediante las condiciones proporcionadas, el cliente puede especificar un valor único o un valor de rango para un tipo de nodo.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
         {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*",
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "StringEquals": {
                    "elasticache:CacheNodeType": [
                        "cache.t2.micro",
                        "cache.t2.medium"
                    ]
                }
            }
        }
    ]
}
```

4. elasticache:CacheNodeType: con Memcached, especifique qué NodeType puede crear un usuario. Mediante las condiciones proporcionadas, el cliente puede especificar un valor único o un valor de rango para un tipo de nodo.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
         {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*"
            ],
            "Condition": {
                "StringEquals": {
                    "elasticache:CacheNodeType": [
                        "cache.t2.micro",
                        "cache.t2.medium"
                    ]
                }
            }
        }
    ]
}
```

5. elasticache:NumNodeGroups: cree un grupo de reproducción con menos de 20 grupos de nodos.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
         {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
             "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
             "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "NumericLessThanEquals": {
                    "elasticache:NumNodeGroups": "20"
                }
            }
        }
    ]
}
```

6. elasticache:ReplicasPerNodeGroup: especifique las réplicas por nodo entre 5 y 10.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
         {
            "Effect": "Allow",
            "Action": [
```

```
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "NumericGreaterThanEquals": {
                    "elasticache:ReplicasPerNodeGroup": "5"
                },
                "NumericLessThanEquals": {
                    "elasticache:ReplicasPerNodeGroup": "10"
                }
            }
        }
    ]
}
```

7. elasticache:EngineVersion: especifique el uso de la versión 5.0.6 del motor.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
     {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
```

```
            ]
        },

        {
           "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*",
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "StringEquals": {
                    "elasticache:EngineVersion": "5.0.6"
                }
            }
        }
    ]
}
```

8. elasticache:EngineVersion: especifique el uso de la versión 1.6.6 del motor de Memcached.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
     {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
```

```
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*"
            ],
            "Condition": {
                "StringEquals": {
                    "elasticache:EngineVersion": "1.6.6"
                }
            }
        }
    ]
}
```

9. elasticache:EngineType: especifique el uso únicamente de un motor de Valkey o Redis OSS.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
         {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*",
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "StringEquals": {
```

```
                    "elasticache:EngineType": "redis"
                }
            }
        }
    ]
}
```

10. elasticache:AtRestEncryptionEnabled: especifique que los grupos de reproducción solo se crearán con el cifrado habilitado.

    JSON

```
{
    "Version":"2012-10-17",
    "Statement": [

         {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "Bool": {
                    "elasticache:AtRestEncryptionEnabled": "true"
                }
            }
        }
    ]
}
```

11. elasticache:TransitEncryptionEnabled

a. Establezca la clave de condición `elasticache:TransitEncryptionEnabled` en `false` para que la acción CreateReplicationGroup especifique que los grupos de replicación solo se puedan crear cuando no se utilice TLS:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "Bool": {
                    "elasticache:TransitEncryptionEnabled": "false"
                }
            }
        }
    ]
}
```

Cuando la clave de condición `elasticache:TransitEncryptionEnabled` esté establecida en `false` en una política para la acción CreateReplicationGroup, solo se permitirá una solicitud `CreateReplicationGroup` si no se utiliza TLS (es decir, si la solicitud no

incluye un parámetro `TransitEncryptionEnabled` establecido en `true` o un parámetro `TransitEncryptionMode` establecido en `required`.

b. Establezca la clave de condición `elasticache:TransitEncryptionEnabled` en `true` para que la acción CreateReplicationGroup especifique que los grupos de replicación solo se puedan crear cuando se utilice TLS:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "Bool": {
                    "elasticache:TransitEncryptionEnabled": "true"
                }
            }
        }
    ]
}
```

Cuando la clave de condición `elasticache:TransitEncryptionEnabled` esté establecida en `true` en una política para la acción CreateReplicationGroup, solo se permitirá una solicitud `CreateReplicationGroup` si la solicitud incluye un

parámetro `TransitEncryptionEnabled` establecido en `true` y un parámetro `TransitEncryptionMode` establecido en `required`.

c. Defina `elasticache:TransitEncryptionEnabled` en `true` para que la acción `ModifyReplicationGroup` especifique que los grupos de replicación solo se pueden modificar cuando se usa TLS:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:ModifyReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "BoolIfExists": {
                    "elasticache:TransitEncryptionEnabled": "true"
                }
            }
        }
    ]
}
```

Cuando la clave de condición `elasticache:TransitEncryptionEnabled` esté establecida en `true` en una política para la acción ModifyReplicationGroup, solo se permitirá una solicitud `ModifyReplicationGroup` si la solicitud incluye un parámetro `TransitEncryptionMode` establecido en `required`. El parámetro `TransitEncryptionEnabled` establecido en `true` también se puede incluir de forma opcional, pero no es necesario en este caso para habilitar TLS.

12. elasticache:AutomaticFailoverEnabled: especifique que los grupos de reproducción solo se crearán con la conmutación por error automática habilitada.

JSON

```
{
    "Version":"2012-10-17",
```

```
    "Statement": [
         {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "Bool": {
                    "elasticache:AutomaticFailoverEnabled": "true"
                }
            }
        }
    ]
}
```

13. elasticache:MultiAZEnabled: especifique que no se pueden crear grupos de reproducción con Multi-AZ deshabilitado.

    JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
         {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
```

```
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },
        {
            "Effect": "Deny",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*",
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "Bool": {
                    "elasticache:MultiAZEnabled": "false"
                }
            }
        }
    ]
}
```

14. elasticache:ClusterModeEnabled: especifique que los grupos de reproducción solo se pueden crear con el modo de clúster habilitado.

    JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
         {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
```

```
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "Bool": {
                    "elasticache:ClusterModeEnabled": "true"
                }
            }
        }
    ]
}
```

15. elasticache:AuthTokenEnabled: especifique que los grupos de reproducción solo se pueden crear con el token AUTH habilitado.

    JSON

```
{
    "Version":"2012-10-17",
    "Statement": [

         {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*",
```

```
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "Bool": {
                    "elasticache:AuthTokenEnabled": "true"
                }
            }
        }
    ]
}
```

16. elasticache:SnapshotRetentionLimit: especifique el número de días (o mín/máx) para mantener la instantánea. A continuación, la política impone el almacenamiento de copias de seguridad durante al menos 30 días.

    JSON

```
{
    "Version":"2012-10-17",
    "Statement": [

         {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup",
                "elasticache:CreateServerlessCache"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*",
                "arn:aws:elasticache:*:*:replicationgroup:*",
                "arn:aws:elasticache:*:*:serverlesscache:*"
```

```
            ],
            "Condition": {
                "NumericGreaterThanEquals": {
                    "elasticache:SnapshotRetentionLimit": "30"
                }
            }
        }
    ]
}
```

17. elasticache:KmsKeyId: especifique el uso de las claves de AWS KMS administradas por el cliente.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
        "Sid": "AllowDependentResources",
        "Effect": "Allow",
        "Action": [
            "elasticache:CreateServerlessCache"
        ],
        "Resource": [
            "arn:aws:elasticache:*:*:serverlesscachesnapshot:*",
            "arn:aws:elasticache:*:*:snapshot:*",
            "arn:aws:elasticache:*:*:usergroup:*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "elasticache:CreateServerlessCache"
        ],
        "Resource": [
            "arn:aws:elasticache:*:*:serverlesscache:*"
        ],
        "Condition": {
            "StringEquals": {
                "elasticache:KmsKeyId": "my-key"
            }
        }
    }
  ]
```

```
}
```

18. elasticache:CacheParameterGroupName: especifique un grupo de parámetros no predeterminado con parámetros específicos de una organización en los clústeres. También puede especificar un patrón de nomenclatura para los grupos de parámetros o eliminar bloques en un nombre de grupo de parámetros específico. Lo que sigue es un ejemplo que restringe el uso de solo “my-org-param-group”.

    JSON

```
{
    "Version":"2012-10-17",
    "Statement": [

         {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*",
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "StringEquals": {
                    "elasticache:CacheParameterGroupName": "my-org-param-group"
                }
            }
        }
    ]
```

```
}
```

19. elasticache:CacheParameterGroupName: con Memcached, especifique un grupo de parámetros no predeterminado con parámetros específicos de una organización en los clústeres. También puede especificar un patrón de nomenclatura para los grupos de parámetros o eliminar bloques en un nombre de grupo de parámetros específico. Lo que sigue es un ejemplo que restringe el uso de solo “my-org-param-group”.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [

         {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*"
            ],
            "Condition": {
                "StringEquals": {
                    "elasticache:CacheParameterGroupName": "my-org-param-group"
                }
            }
        }
    ]
}
```

20. elasticache:CreateCacheCluster: denegar la acción `CreateCacheCluster` si falta la etiqueta de solicitud `Project` o no es igual a `Dev`, `QA` o `Prod`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
          {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*",
                "arn:aws:elasticache:*:*:securitygroup:*",
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ]
        },
        {
            "Effect": "Deny",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*"
            ],
            "Condition": {
                "Null": {
                    "aws:RequestTag/Project": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:AddTagsToResource"
            ],
            "Resource": "arn:aws:elasticache:*:*:cluster:*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/Project": [
```

```
                    "Dev",
                    "Prod",
                    "QA"
                ]
            }
        }
    ]
}
```

21. elasticache:CacheNodeType: permite `CreateCacheCluster` con `cacheNodeType` cache.r5.large o cache.r6g.4xlarge y la etiqueta `Project=XYZ`.

    JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
      {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
      ],
      "Condition": {
        "StringEqualsIfExists": {
          "elasticache:CacheNodeType": [
            "cache.r5.large",
            "cache.r6g.4xlarge"
          ]
        },
```

```
        "StringEquals": {
          "aws:RequestTag/Project": "XYZ"
        }
      }
    }
  ]
}
```

22. elasticache:CacheNodeType: permite `CreateCacheCluster` con `cacheNodeType` cache.r5.large o cache.r6g.4xlarge y la etiqueta `Project=XYZ`.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
      {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
      ],
      "Condition": {
        "StringEqualsIfExists": {
          "elasticache:CacheNodeType": [
            "cache.r5.large",
            "cache.r6g.4xlarge"
          ]
        },
        "StringEquals": {
          "aws:RequestTag/Project": "XYZ"
        }
```

```
      }
    }
  ]
}
```

> **Note**
>
> Cuando se crean políticas a fin de imponer etiquetas y otras claves de condición juntas, el condicional `IfExists` puede ser necesario en los elementos de la clave de condición debido a los requisitos extra de la política `elasticache:AddTagsToResource` para las solicitudes de creación con el parámetro `--tags`.

Uso de roles vinculados a servicios para Amazon ElastiCache

Amazon ElastiCache utiliza roles vinculados a servicios de AWS Identity and Access Management (IAM). Un rol vinculado a servicios es un tipo único de rol de IAM que se encuentra vinculado directamente a un servicio de AWS, como Amazon ElastiCache. Amazon ElastiCache predefine los roles vinculados a servicios de Amazon ElastiCache. Incluyen todos los permisos que requiere el servicio para llamar a otros servicios de AWS en nombre de los clústeres.

Un rol vinculado a servicios simplifica la configuración de Amazon ElastiCache porque ya no tendrá que agregar manualmente los permisos necesarios. Los roles ya existen dentro de la cuenta de AWS, pero se encuentran vinculados a los casos de uso de Amazon ElastiCache y tienen permisos predefinidos. Solo Amazon ElastiCache puede asumir estos roles y solo estos roles pueden utilizar la política de permisos predefinida. Las funciones se pueden eliminar únicamente después de eliminar primero sus recursos relacionados. De esta forma se protegen los recursos de Amazon ElastiCache porque no puede eliminar los permisos necesarios para acceder a los recursos.

Para obtener información acerca de otros servicios que son compatibles con roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM y busque los servicios que muestran Sí en la columna Rol vinculado a un servicio. Elija una opción Sí con un enlace para ver la documentación acerca del rol vinculado a servicios en cuestión.

Contenido

- Permisos de roles vinculados a servicios para Amazon ElastiCache
  - Permisos para crear un rol vinculado a servicios

- Creación de un rol vinculado a servicios (IAM)
  - Creación de un rol vinculado a servicios (consola de IAM)
  - Creación de un rol vinculado a servicios (CLI de IAM)
  - Creación de un rol vinculado a servicios (API de IAM)
- Edición de la descripción de un rol vinculado a servicios para Amazon ElastiCache
  - Edición de la descripción de un rol vinculado a servicios (consola de IAM)
  - Edición de la descripción de un rol vinculado a servicios (CLI de IAM)
  - Edición de la descripción de un rol vinculado a servicios (API de IAM)
- Eliminar un rol vinculado a servicios para Amazon ElastiCache
  - Limpiar un rol vinculado a un servicio
  - Eliminación de un rol vinculado a servicios (consola de IAM)
  - Eliminación de un rol vinculado a servicios (CLI de IAM)
  - Eliminación de un rol vinculado a servicios (API de IAM)

Permisos de roles vinculados a servicios para Amazon ElastiCache

Permisos para crear un rol vinculado a servicios

Para permitir a una entidad de IAM crear roles vinculados a servicios AWSServiceRoleForElastiCache

Agregue la siguiente instrucción de política a los permisos para esa entidad de IAM:

```
{
    "Effect": "Allow",
    "Action": [
        "iam:CreateServiceLinkedRole",
        "iam:PutRolePolicy"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/elasticache.amazonaws.com/
AWSServiceRoleForElastiCache*",
    "Condition": {"StringLike": {"iam:AWSServiceName": "elasticache.amazonaws.com"}}
}
```

Para permitir a una entidad de IAM eliminar roles vinculados a servicios AWSServiceRoleForElastiCache

Agregue la siguiente instrucción de política a los permisos para esa entidad de IAM:

```
{
    "Effect": "Allow",
    "Action": [
        "iam:DeleteServiceLinkedRole",
        "iam:GetServiceLinkedRoleDeletionStatus"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/elasticache.amazonaws.com/
AWSServiceRoleForElastiCache*",
    "Condition": {"StringLike": {"iam:AWSServiceName": "elasticache.amazonaws.com"}}
}
```

También puede utilizar una política administrada por AWS para proporcionar acceso completo a Amazon ElastiCache.

Creación de un rol vinculado a servicios (IAM)

Puede crear un rol vinculado a servicios mediante la consola de IAM, la CLI o la API.

Creación de un rol vinculado a servicios (consola de IAM)

Puede utilizar la consola de IAM para crear un rol vinculado a un servicio.

Para crear un rol vinculado a un servicio (consola)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación de la consola de IAM, elija Roles. A continuación, elija Create new role (Crear nuevo rol).
3. En Select type of trusted entity (Seleccionar el tipo de entidad de confianza), elija AWS Service (Servicio de).
4. En Or select a service to view its use cases (O seleccionar un servicio para ver sus casos de uso), elija ElastiCache.
5. Elija Siguiente: permisos.
6. En Policy name (Nombre de la política), tenga en cuenta que `ElastiCacheServiceRolePolicy` es necesario para este rol. Elija Siguiente:Etiquetas.
7. Tenga en cuenta que las etiquetas no son compatibles con los roles vinculados a servicios. Elija Next: Review.

8. (Opcional) En Descripción del rol, edite la descripción del nuevo rol vinculado al servicio.
9. Revise el rol y, a continuación, seleccione Crear rol.

Creación de un rol vinculado a servicios (CLI de IAM)

Puede utilizar las operaciones de IAM desde la AWS Command Line Interface para crear un rol vinculado a servicios. Este rol puede incluir la política de confianza y las políticas insertadas que el servicio necesita para asumir el rol.

Para crear un rol vinculado a un servicio (CLI)

Use la operación siguiente:

```
$ aws iam create-service-linked-role --aws-service-name elasticache.amazonaws.com
```

Creación de un rol vinculado a servicios (API de IAM)

Puede utilizar la API de IAM para crear un rol vinculado a servicios. Este rol puede contener la política de confianza y las políticas insertadas que el servicio necesita para asumir el rol.

Para crear un rol vinculado a un servicio (API)

Use la llamada de API de CreateServiceLinkedRole. En la solicitud, especifique el nombre del servicio de `elasticache.amazonaws.com`.

Edición de la descripción de un rol vinculado a servicios para Amazon ElastiCache

Amazon ElastiCache no permite editar el rol vinculado a servicios de AWSServiceRoleForElastiCache. Después de crear un rol vinculado al servicio, no podrá cambiar el nombre del rol, ya que varias entidades podrían hacer referencia al rol. Sin embargo, sí puede editar la descripción del rol con IAM.

Edición de la descripción de un rol vinculado a servicios (consola de IAM)

Puede utilizar la consola de IAM para editar una descripción de rol vinculado a servicios.

Para editar la descripción de un rol vinculado a un servicio (consola)

1. En el panel de navegación de la consola de IAM, elija Roles.
2. Seleccione el nombre del rol que desea modificar.

3. En el extremo derecho de Role description, seleccione Edit.
4. Ingrese una descripción nueva en el cuadro Save (Guardar).

Edición de la descripción de un rol vinculado a servicios (CLI de IAM)

Puede utilizar operaciones de IAM desde la AWS Command Line Interface para editar una descripción de rol vinculado a servicios.

Para cambiar la descripción de un rol vinculado a un servicio (CLI)

1. (Opcional) A fin de ver la descripción actual de un rol, utilice la AWS CLI para la operación de IAM `get-role`.

   Example

   ```
   $ aws iam get-role --role-name AWSServiceRoleForElastiCache
   ```

   Utilice el nombre del rol, no el ARN, para hacer referencia a los roles con las operaciones de la CLI. Por ejemplo, si una función tiene el ARN `arn:aws:iam::123456789012:role/myrole`, debe referirse a él como **`myrole`**.
2. Para actualizar la descripción de un rol vinculado a servicios, utilice la AWS CLI para la operación de IAM `update-role-description`.

   Para Linux, macOS o Unix:

   ```
   $ aws iam update-role-description \
       --role-name AWSServiceRoleForElastiCache \
       --description "new description"
   ```

   Para Windows:

   ```
   $ aws iam update-role-description ^
       --role-name AWSServiceRoleForElastiCache ^
       --description "new description"
   ```

Edición de la descripción de un rol vinculado a servicios (API de IAM)

Puede utilizar la API de IAM para editar una descripción de rol vinculado a servicios.

Para cambiar la descripción de un rol vinculado a un servicio (API)

1. (Opcional) Para ver la descripción actual de un rol, utilice la operación de la API de IAM GetRole.

   Example

   ```
   https://iam.amazonaws.com/
      ?Action=GetRole
      &RoleName=AWSServiceRoleForElastiCache
      &Version=2010-05-08
      &AUTHPARAMS
   ```

2. Para actualizar la descripción de un rol, utilice la operación de la API de IAM UpdateRoleDescription.

   Example

   ```
   https://iam.amazonaws.com/
      ?Action=UpdateRoleDescription
      &RoleName=AWSServiceRoleForElastiCache
      &Version=2010-05-08
      &Description="New description"
   ```

Eliminar un rol vinculado a servicios para Amazon ElastiCache

Si ya no necesita utilizar una característica o servicio que requiere un rol vinculado a un servicio, recomendamos que elimine dicho rol. De esta forma no tiene una entidad no utilizada que no se monitoree ni mantenga de forma activa. Sin embargo, debe limpiar el rol vinculado al servicio antes de eliminarlo.

Amazon ElastiCache no elimina de forma automática el rol vinculado a servicios.

Limpiar un rol vinculado a un servicio

Antes de que pueda utilizar IAM para eliminar un rol vinculado a servicios, primero confirme que el rol no tiene recursos (clústeres o grupos de reproducción) asociados a él.

Para comprobar si el rol vinculado a un servicio tiene una sesión activa en la consola de IAM

1. Inicie sesión en la Consola de administración de AWS y abra la consola de IAM en https://console.aws.amazon.com/iam/.

2. En el panel de navegación de la consola de IAM, elija Roles. A continuación, elija el nombre (no la casilla de verificación) del rol AWSServiceRoleForElastiCache.
3. En la página Resumen del rol seleccionado, seleccione la pestaña Asesor de acceso.
4. En la pestaña Asesor de acceso, revise la actividad reciente del rol vinculado a servicios.

Para eliminar recursos de Amazon ElastiCache que requieren AWSServiceRoleForelastiCache

- Para eliminar un clúster, consulte los siguientes temas:
  - Uso de Consola de administración de AWS
  - Uso de AWS CLI para eliminar un clúster de ElastiCache
  - Uso de la API de ElastiCache
- Para eliminar un grupo de reproducción, consulte los siguientes temas:
  - Eliminación de un grupo de reproducción (consola)
  - Eliminación de un grupo de reproducción de (AWS CLI)
  - Eliminación de un grupo de reproducción (API de ElastiCache)

Eliminación de un rol vinculado a servicios (consola de IAM)

Puede utilizar la consola de IAM para eliminar un rol vinculado a un servicio.

Para eliminar un rol vinculado a un servicio (consola)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación de la consola de IAM, elija Roles. A continuación, seleccione la casilla junto al nombre del rol que desea eliminar, no el nombre ni la fila.
3. En Role actions (Acciones de rol) en la parte superior de la página, elija Delete role (Eliminar rol).
4. En el cuadro de diálogo de confirmación, revise los datos del último acceso al servicio, que muestra cuándo cada una de las funciones seleccionadas tuvo acceso a un servicio de AWS por última vez. Esto lo ayuda a confirmar si el rol está actualmente activo. Si desea continuar, seleccione Yes, Delete para enviar la solicitud de eliminación del rol vinculado al servicio.
5. Consulte las notificaciones de la consola de IAM para monitorear el progreso de la eliminación del rol vinculado al servicio. Como el proceso de eliminación del rol vinculado al servicio de IAM es asíncrono, dicha tarea puede realizarse correctamente o fallar después de que envía

la solicitud de eliminación. Si la tarea no se realiza correctamente, puede seleccionar View details (Ver detalles) o View Resources (Ver recursos) desde las notificaciones para obtener información sobre el motivo por el que no se pudo eliminar el rol.

Eliminación de un rol vinculado a servicios (CLI de IAM)

Puede utilizar las operaciones de IAM desde la AWS Command Line Interface para eliminar un rol vinculado a servicios.

Para eliminar un rol vinculado a un servicio (CLI)

1. Si no conoce el nombre del rol vinculado a servicios que desea eliminar, ingrese el siguiente comando. En este comando se enumeran los roles y los nombres de recursos de Amazon (ARN) de la cuenta.

   ```
   $ aws iam get-role --role-name role-name
   ```

   Utilice el nombre del rol, no el ARN, para hacer referencia a los roles con las operaciones de la CLI. Por ejemplo, si un rol tiene el ARN `arn:aws:iam::123456789012:role/myrole`, debe referirse a él como **`myrole`**.

2. Como los roles vinculados a servicios no se puede eliminar si están en uso o tienen recursos asociados, debe enviar una solicitud de eliminación. Esta solicitud puede denegarse si no se cumplen estas condiciones. Debe apuntar el valor `deletion-task-id` de la respuesta para comprobar el estado de la tarea de eliminación. Ingrese lo siguiente para enviar una solicitud de eliminación de un rol vinculado a servicios.

   ```
   $ aws iam delete-service-linked-role --role-name role-name
   ```

3. Ingrese lo siguiente para verificar el estado de la tarea de eliminación.

   ```
   $ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
   ```

   El estado de la tarea de eliminación puede ser `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` o `FAILED`. Si ocurre un error durante la eliminación, la llamada devuelve el motivo del error para que pueda resolver el problema.

Eliminación de un rol vinculado a servicios (API de IAM)

Puede utilizar la API de IAM para eliminar un rol vinculado a un servicio.

Para eliminar un rol vinculado a un servicio (API)

1. Para enviar una solicitud de eliminación de un rol vinculado a un servicio, llame a DeleteServiceLinkedRole. En la solicitud, especifique el nombre del rol.

   Como los roles vinculados a servicios no se puede eliminar si están en uso o tienen recursos asociados, debe enviar una solicitud de eliminación. Esta solicitud puede denegarse si no se cumplen estas condiciones. Debe apuntar el valor `DeletionTaskId` de la respuesta para comprobar el estado de la tarea de eliminación.
2. Para comprobar el estado de la tarea de eliminación, realice una llamada a GetServiceLinkedRoleDeletionStatus. En la solicitud, especifique el valor de `DeletionTaskId`.

   El estado de la tarea de eliminación puede ser `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` o `FAILED`. Si ocurre un error durante la eliminación, la llamada devuelve el motivo del error para que pueda resolver el problema.

Permisos de la API de ElastiCache: referencia de condiciones, acciones y recursos

Cuando configure el control de acceso y escriba políticas de permisos para adjuntar a una política de IAM (políticas basadas en identidad o recurso), utilice la siguiente tabla como referencia. En la tabla se enumera cada operación de la API de Amazon ElastiCache y las acciones correspondientes para las que puede conceder permisos a fin de realizar la acción. Las acciones se especifican en el campo `Action` de la política y el valor de un recurso se especifica en el campo `Resource` de la política. A menos que se indique lo contrario, el recurso es necesario. Algunos campos incluyen recursos obligatorios y opcionales. Cuando no hay ARN de recurso, el recurso de la política es un comodín (*).

Puede utilizar claves de condiciones en las políticas de ElastiCache para expresar condiciones. Para ver una lista de claves de condición específicas de Elasticache, junto con las acciones y los tipos de recursos a los que se aplican, consulte Uso de claves de condición. Para ver una lista completa de claves generales de AWS, consulte Claves de contexto de condición globales de AWS en la Guía del usuario de IAM.

> **Note**
>
> Para especificar una acción, use el prefijo `elasticache:` seguido del nombre de operación de la API (por ejemplo, `elasticache:DescribeCacheClusters`).

Para ver una lista de las acciones de ElastiCache, consulte Acciones definidas por Amazon ElastiCache, en la Referencia de autorizaciones de servicio.

Validación de la conformidad para Amazon ElastiCache

Auditores externos evalúan la seguridad y la conformidad de los servicios de AWS como parte de varios programas de conformidad de AWS, como SOC, PCI, FedRAMP e HIPAA.

Para saber si un Servicio de AWS está incluido en el ámbito de programas de conformidad específicos, consulte Servicios de AWS incluidos por programa de conformidad y escoja el programa de conformidad que le interese. Para obtener información general, consulte Programas de conformidad de AWS.

Puedes descargar los informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact.

Su responsabilidad de conformidad al utilizar Servicios de AWS se determina en función de la confidencialidad de los datos, los objetivos de conformidad de su empresa, así como de la legislación y los reglamentos aplicables. Para obtener más información sobre la responsabilidad de conformidad al usar Servicios de AWS, consulte la Documentación de seguridad de AWS.

- Cuando Amazon ElastiCache aparece dentro del ámbito de un programa de conformidad, este ámbito se aplica a todos los motores de caché compatibles con ElastiCache. Estos motores son ElastiCache para Valkey, ElastiCache para Memcached y ElastiCache para Redis OSS.

Más información

Para obtener información general sobre la conformidad de la nube de AWS, consulte lo siguiente:

- Puntos de conexión FIPS por servicio
- Actualizaciones de servicios en ElastiCache
- AWS Conformidad en la nube
- Modelo de responsabilidad compartida
- AWS Programa de conformidad con PCI DSS de

Resiliencia en Amazon ElastiCache

La infraestructura global de AWS se compone de regiones y zonas de disponibilidad de AWS. AWS Las regiones proporcionan varias zonas de disponibilidad físicamente independientes y aisladas que se encuentran conectadas mediante redes con un alto nivel de rendimiento y redundancia, además de baja latencia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre zonas de disponibilidad sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de centros de datos únicos o múltiples.

Para obtener más información sobre las regiones y zonas de disponibilidad de AWS, consulte Infraestructura global de AWS.

Además de la infraestructura global de AWS, Amazon ElastiCache ofrece varias características que ayudan con las necesidades de resiliencia y copia de seguridad de los datos.

Temas

- Mitigación de errores

Mitigación de errores

A la hora de planificar la implementación de Amazon ElastiCache, debe realizar la planificación de modo que los errores tengan una repercusión mínima en la aplicación y datos. Los temas de esta sección abordan enfoques que puede aplicar para proteger la aplicación y los datos frente a errores.

Temas

- Mitigación de errores al ejecutar Memcached
- Mitigación de errores al ejecutar Valkey o Redis OSS
- Recomendaciones

Mitigación de errores al ejecutar Memcached

Al ejecutar el motor de Memcached, dispondrá de las opciones siguientes para minimizar el impacto de los errores. Existen dos tipos de errores que debe abordar en los planes de mitigación: errores de nodos y errores de zonas de disponibilidad.

Mitigación de errores de nodos

Las cachés sin servidor mitigan automáticamente los errores de los nodos con una arquitectura replicada de varias zonas de disponibilidad para que los errores de los nodos sean transparentes para su aplicación. Para mitigar el impacto de un error de nodo en un clúster basado en nodos, reparta los datos almacenados en caché por varios nodos. Como los clústeres basados en nodos no son compatibles con la replicación, los errores en nodos siempre tendrán como consecuencia la pérdida de datos del clúster.

Al crear su clúster de Memcached, puede crearlo con un número de nodos comprendido entre 1 y 60, o más si se solicita de forma especial. Repartir los datos entre un mayor número de nodos implica que perderá menos datos en caso de error en algún nodo. Por ejemplo, si reparte los datos en 10 nodos, un único nodo almacenará aproximadamente un 10 % de los datos almacenados en la caché. En este caso, un error de nodo supondrá una pérdida aproximada del 10 % de la caché, que deberá reemplazarse cuando se cree y aprovisione un nodo de reemplazo. Si los datos se almacenan en caché en 3 nodos de mayor tamaño, un error en un nodo supondría una pérdida aproximada del 33 % de los datos almacenados en la caché.

Para obtener información acerca de la especificación del número de nodos en un clúster de Memcached, consulte Creación de un clúster de Memcached (consola).

Mitigación de errores de zona de disponibilidad

Las cachés sin servidor mitigan automáticamente los errores de las zonas de disponibilidad con una arquitectura replicada de varias zonas de disponibilidad a fin de que los errores de estas zonas sean transparentes para su aplicación.

Para mitigar el impacto de los fallos de una zona de disponibilidad en un clúster basado en nodos, busque los nodos en tantas zonas de disponibilidad como sea posible. En el caso improbable de que se produzca un error en una zona de disponibilidad, se perderán los datos almacenados en la caché de dicha zona, no los datos almacenados en la caché de otras zonas de disponibilidad.

¿Por qué tantos nodos?

Dado que mi región solo tiene tres zonas de disponibilidad, si se produjera un error en una de ellas, perdería aproximadamente un tercio de los datos. En ese caso, ¿por qué son necesarios más de tres nodos?

Esta es una excelente pregunta. Recuerde que estamos intentando mitigar dos tipos distintos de errores: errores de nodos y errores de zonas de disponibilidad. Tiene razón. Si los datos están distribuidos en distintas zonas de disponibilidad y, si se produce un error en una de ellas, solo perderá los datos almacenados en la memoria caché de dicha zona de disponibilidad, independientemente del número de nodos que tenga. Sin embargo, en caso de error en un nodo, disponer de más nodos reducirá la proporción de datos perdidos.

No existe ninguna “fórmula mágica” para determinar la cantidad de nodos que debe tener en un clúster. Debe sopesar el impacto de la pérdida de datos frente a la probabilidad de que se produzca algún error y los costos para llegar a su propia conclusión.

Para obtener información acerca de la especificación del número de nodos en un clúster de Memcached, consulte Creación de un clúster de Memcached (consola).

Para obtener más información acerca de las regiones y zonas de disponibilidad, consulte Elección de regiones y zonas de disponibilidad para ElastiCache.

Mitigación de errores al ejecutar Valkey o Redis OSS

Si ejecuta un motor de Valkey o Redis OSS, dispondrá de las siguientes opciones para minimizar el impacto de los errores que se produzcan en un nodo o zona de disponibilidad.

Mitigación de errores de nodos

Las cachés sin servidor mitigan automáticamente los errores de los nodos con una arquitectura de varias zonas de disponibilidad a fin de que los errores de los nodos sean transparentes para su aplicación. Los clústeres basados en nodos deben configurarse adecuadamente para mitigar el fallo de un nodo individual.

Para mitigar el impacto de los errores de los nodos de Valkey o Redis OSS en los clústeres basados en nodos, dispone de las siguientes opciones:

Temas

- Mitigación de errores: grupos de replicación de Valkey o Redis OSS

Mitigación de errores: grupos de replicación de Valkey o Redis OSS

Un grupo de replicación de Valkey o Redis OSS se compone de un solo nodo principal, disponible para operaciones de lectura y escritura para su aplicación, y de 1 a 5 nodos de réplica de solo lectura. Cuando se escriben datos en el nodo principal, también se actualizan de forma asíncrona en los nodos de réplica de lectura.

Qué sucede en caso de error en una réplica de lectura

1. ElastiCache detecta la réplica de lectura con error.
2. ElastiCache deja el nodo con error sin conexión.
3. ElastiCache lanza y aprovisiona un nodo de reemplazo en la misma zona de disponibilidad.
4. El nuevo nodo se sincroniza con el nodo principal.

Durante este tiempo, la aplicación podrá seguir realizando operaciones de lectura y escritura con los demás nodos.

Multi-AZ de Valkey o Redis OSS

Puede habilitar multi-AZ en sus grupos de replicación de Valkey o Redis OSS. Independientemente de que habilite Multi-AZ o no, se detectará y reemplazará automáticamente un error en el nodo principal. El modo en que esto tiene lugar varía en función de si Multi-AZ está habilitado o no.

Cuando Multi-AZ está habilitado

1. ElastiCache detecta el error del nodo primario.

2. ElastiCache promociona el nodo de réplica de lectura con el mínimo retraso de reproducción en el nodo primario.
3. El resto de réplicas se sincronizarán con el nuevo nodo principal.
4. ElastiCache pone en marcha una réplica de lectura en la zona de disponibilidad del nodo primario con error.
5. El nuevo nodo se sincroniza con el nodo principal recién promovido.

La conmutación por error a un nodo de réplica suele ser más rápida que la creación y el aprovisionamiento de un nuevo nodo principal. Esto significa que la aplicación podrá reanudar la escritura en el nodo principal antes que si Multi-AZ no estuviera habilitado.

Para obtener más información, consulte Minimización del tiempo de inactividad en ElastiCache utilizando multi-AZ con Valkey y Redis OSS.

Cuando Multi-AZ está deshabilitado

1. ElastiCache detecta un error primario.
2. ElastiCache deja el primario sin conexión.
3. ElastiCache crea y aprovisiona un nodo primario nuevo para reemplazar el primario con error.
4. ElastiCache sincroniza el nodo primario nuevo con una de las réplicas existentes.
5. Cuando finaliza la sincronización, el nuevo nodo funciona como el nodo principal del clúster.

Durante los pasos 1 a 4 de este proceso, la aplicación no puede escribir en el nodo principal. Sin embargo, la aplicación podrá seguir leyendo datos de los nodos de réplica.

Para mejorar la protección, le recomendamos que lance los nodos del grupo de reproducción en distintas zonas de disponibilidad (AZ). De este modo, los errores en zonas de disponibilidad solo afectarán a los nodos de dichas zonas de disponibilidad, no a los demás.

Para obtener más información, consulte Alta disponibilidad a través de grupos de reproducción.

Mitigación de errores de zona de disponibilidad

Las cachés sin servidor mitigan automáticamente los errores de las zonas de disponibilidad con una arquitectura replicada de varias zonas de disponibilidad a fin de que los errores de estas zonas sean transparentes para su aplicación.

Para mitigar el impacto de un fallo de una zona de disponibilidad en un clúster basado en nodos, busque los nodos de cada partición en tantas zonas de disponibilidad como sea posible.

Independientemente de la cantidad de nodos que tenga en una partición, si se encuentran en la misma zona de disponibilidad, un error catastrófico en dicha zona tendría como resultado la pérdida de todos los datos de la partición. Sin embargo, si ubica los nodos en varias zonas de disponibilidad, un error en cualquiera de las zonas de disponibilidad tendría como consecuencia la pérdida solo de los nodos de dicha zona de disponibilidad.

Cada vez que se pierde un nodo, puede experimentar una reducción del rendimiento, ya que las operaciones de lectura son compartidas por menos nodos. Esta reducción del rendimiento continuará hasta que los nodos se reemplacen.

Para obtener más información acerca de la especificación de zonas de disponibilidad en los nodos de Valkey o Redis OSS, consulte Creación de un clúster de Valkey (modo de clúster deshabilitado) (consola).

Para obtener más información acerca de las regiones y zonas de disponibilidad, consulte Elección de regiones y zonas de disponibilidad para ElastiCache.

Recomendaciones

Se recomienda crear cachés sin servidor en clústeres basados en nodos, ya que obtendrá automáticamente una mejor tolerancia a errores sin necesidad de configuración adicional. Sin embargo, al crear un clúster basado en nodos, hay dos tipos de fallos para los que se debe preparar: los fallos de nodos individuales y los fallos generalizados en zonas de disponibilidad. Los mejores planes de mitigación de errores abordarán ambos tipos de errores.

Minimización del impacto de los errores de nodos

Para minimizar el impacto de un error en un nodo al utilizar Valkey o Redis OSS, recomendamos que su implementación utilice varios nodos en cada partición y distribuya los nodos en varias zonas de disponibilidad. Esto se hace automáticamente en las cachés sin servidor.

En los clústeres basados en nodos en Valkey o Redis OSS, se recomienda habilitar multi-AZ en el grupo de replicación para que ElastiCache conmute por error de forma automática a una réplica si falla el nodo principal.

Cuando ejecute Memcached y tenga distribuidos los datos en distintos nodos, cuantos más nodos utilice, menor será la pérdida de datos en caso de error en algún nodo.

Minimización del impacto de los errores en la zona de disponibilidad

Para minimizar el impacto de un error en una zona de disponibilidad, recomendamos lanzar los nodos en tantas zonas de disponibilidad distintas como sea posible. Distribuir los nodos de manera regular en zonas de disponibilidad minimizará el impacto en caso de un improbable evento de error de zona de disponibilidad. Esto se hace automáticamente en las cachés sin servidor.

Otras precauciones

Si ejecuta Valkey o Redis OSS, además de todo lo anterior, recomendamos que programe copias de seguridad periódicas del clúster. Las copias de seguridad (instantáneas) crean un archivo .rdb que podrá utilizar para restablecer la caché en caso de error o daño. Para obtener más información, consulte Instantánea y restauración.

Seguridad de la infraestructura en ElastiCache de AWS

Como servicio administrado, ElastiCache de AWS se encuentra protegido por los procedimientos de seguridad de red globales de AWS que se describen en la sección de Seguridad y conformidad, en el Centro de arquitectura de AWS.

Puede utilizar llamadas a la API publicadas en AWS para obtener acceso a ElastiCache a través de la red. Los clientes deben ser compatibles con Transport Layer Security (TLS) 1.2 o una versión posterior. Recomendamos TLS 1.3 o una versión posterior. Los clientes también deben ser compatibles con conjuntos de cifrado con confidencialidad directa total (PFS) tales como Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad principal de IAM. También puedes utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Actualizaciones de servicios en ElastiCache

ElastiCache supervisa automáticamente la flota de cachés, clústeres y nodos a fin de aplicar actualizaciones de servicio cuando estén disponibles. Las actualizaciones de servicio para las cachés sin servidor se aplican de forma transparente y automática. En los clústeres basados en nodos, se configura un periodo de mantenimiento predefinido para que ElastiCache pueda aplicar

las actualizaciones. Sin embargo, en algunos casos es posible que este enfoque resulte demasiado rígido y que probablemente restrinja los flujos de negocio.

Con las actualizaciones de servicio, usted decide qué actualizaciones se implementan en sus clústeres basados en nodos y cuándo. También puede supervisar en tiempo real el progreso de estas actualizaciones en los clústeres de ElastiCache que usted seleccione.

Temas

- Administración de actualizaciones de servicio para clústeres basados en nodos

Administración de actualizaciones de servicio para clústeres basados en nodos

Las actualizaciones de servicio de ElastiCache para clústeres basados en nodos se publican de forma periódica. Si tiene uno o varios clústeres basados en nodos que reúnen los requisitos de estas actualizaciones, recibirá notificaciones por correo electrónico, SNS, Personal Health Dashboard (PHD) y eventos de Amazon CloudWatch cuando se lanzan las actualizaciones. Las actualizaciones se muestran también en la página de Service Updates (Actualización de servicio) de la consola de ElastiCache. Mediante este panel, puede consultar todas las actualizaciones de servicio y su estado para la flota de ElastiCache. Las actualizaciones de servicio para las cachés sin servidor se aplican de forma transparente y no se pueden administrar mediante actualizaciones de servicio.

Puede controlar cuándo se debe aplicar una actualización antes de que se inicie una actualización automática. Es extremadamente importante que aplique las actualizaciones del tipo security-update para asegurarse de que los clústeres de ElastiCache estén siempre actualizados con los parches de seguridad actuales.

En las siguientes secciones se describen detalladamente las opciones.

Aplicación de las actualizaciones de servicio

Puede comenzar a aplicar las actualizaciones del servicio a la flota Redis desde el momento en que las actualizaciones tengan el estado available (disponible). Las actualizaciones del servicio son acumulativas. Es decir, todas las actualizaciones que no se hayan aplicado se incluirán con la última actualización.

Si una actualización de servicio tiene habilitada la actualización automática, puede optar por no realizar ninguna acción cuando esté disponible. ElastiCache programará la aplicación de la

actualización durante el período de mantenimiento de los clústeres después de la Fecha de inicio de la actualización automática. Recibirá notificaciones relacionadas con cada etapa de la actualización.

> **Note**
>
> Solo puede aplicar las actualizaciones de servicio que tengan un estado disponible o programado.

Para obtener más información sobre cómo revisar y aplicar actualizaciones específicas del servicio a los clústeres de ElastiCache correspondientes, consulte Aplicación de las actualizaciones de servicio con la consola.

Cuando una actualización de servicio nueva se encuentra disponible para uno o más clústeres de ElastiCache, puede utilizar la consola de ElastiCache, la API o la AWS CLI a fin de aplicar la actualización. En las siguientes secciones se explican las opciones que puede utilizar para aplicar las actualizaciones.

Aplicación de las actualizaciones de servicio con la consola

Para consultar la lista de las distintas actualizaciones de servicio disponibles, junto con otra información, vaya a Service Updates (Actualizaciones de servicio) en la consola.

1. Inicie sesión en la Consola de administración de AWS y abra la consola de Amazon ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, seleccione Service Updates (Actualizaciones de servicio).
3. En Service Updates (Actualizaciones de servicio) puede consultar lo siguiente:
   - Service update name (Nombre de actualización de servicio): el nombre único de la actualización de servicio
   - Update type (Tipo de actualización): el tipo de actualización de servicio, que es o security-update o engine-update
   - Update Severity (Gravedad de la actualización): la prioridad de aplicación de la actualización:
     - critical (crítica): recomendamos que aplique esta actualización inmediatamente (en un plazo de 14 días o menos).
     - important (importante): recomendamos que aplique esta actualización tan pronto como el flujo de negocio lo permita (en un plazo de 30 días o menos).

- medium (media): recomendamos que aplique esta actualización tan pronto como pueda (en un plazo de 60 días o menos).
- low (baja): recomendamos que aplique esta actualización tan pronto como pueda (en un plazo de 90 días o menos).

- Engine version (Versión del motor): si el tipo de actualización es actualización del motor, la versión del motor que se está actualizando.
- Fecha de lanzamiento: cuándo se publica la versión y cuándo está disponible para su aplicación en los clústeres.
- Fecha de aplicación recomendada: fecha en la que ElastiCache recomienda aplicar las actualizaciones.
- Status (Estado): el estado de la actualización, que es uno de los siguientes:
  - disponible: la aplicación está lista para los clústeres correspondientes.
  - complete (completa): la actualización se ha aplicado correctamente.
  - cancelled (cancelada): la actualización se ha cancelado y ya no es necesaria.
  - expired (caducada): la actualización ya no está disponible para ser aplicada.

4. Elija una actualización individual (no el botón de la izquierda) para ver los detalles de la actualización del servicio.

   En la sección Cluster update status (Estado de actualización del clúster), puede ver una lista de clústeres en los que la actualización del servicio no se ha aplicado o se ha aplicado recientemente. Para cada clúster, puede ver lo siguiente:

   - Cluster name (Nombre del clúster): el nombre del clúster
   - Nodes Updated (Nodos actualizados): la proporción de nodos en un clúster específico que se actualizaron o que permanecen disponibles para la actualización del servicio específica.
   - Update Type (Tipo de actualización): el tipo de actualización de servicio, que es security-update o engine-update
   - Status (Estado): el estado de la actualización de servicio en el clúster, que es uno de los siguientes:
     - available (disponible): la aplicación está lista para los clústeres Redis correspondientes.
     - en progreso: la actualización se está aplicación a este clúster.
     - scheduled (programado): se ha programado la fecha de actualización.
     - complete (completa): la actualización se ha aplicado correctamente. El clúster con el estado completo se mostrará durante 7 días después de su finalización.

Si ha elegido alguno o todos los clústeres con estado available (disponible) o scheduled (programado) y, luego, eligió Apply now (Postúlese ahora), la actualización empezará a aplicarse en esos clústeres.

Aplicación de las actualizaciones de servicio con la AWS CLI

Tras recibir una notificación de que hay actualizaciones del servicio disponibles, puede inspeccionarlas y aplicarlas con AWS CLI:

- Para recuperar una descripción de las actualizaciones de servicio disponibles, ejecute el siguiente comando:

  ```
  aws elasticache describe-service-updates --service-update-status
  available
  ```

  Para obtener más información, consulte describe-service-updates.
- Para aplicar una actualización de servicio en una lista de clústeres, ejecute el siguiente comando:

  ```
  aws elasticache batch-apply-update-action --service-update
  ServiceUpdateNameToApply=sample-service-update --cluster-names cluster-1
  cluster2
  ```

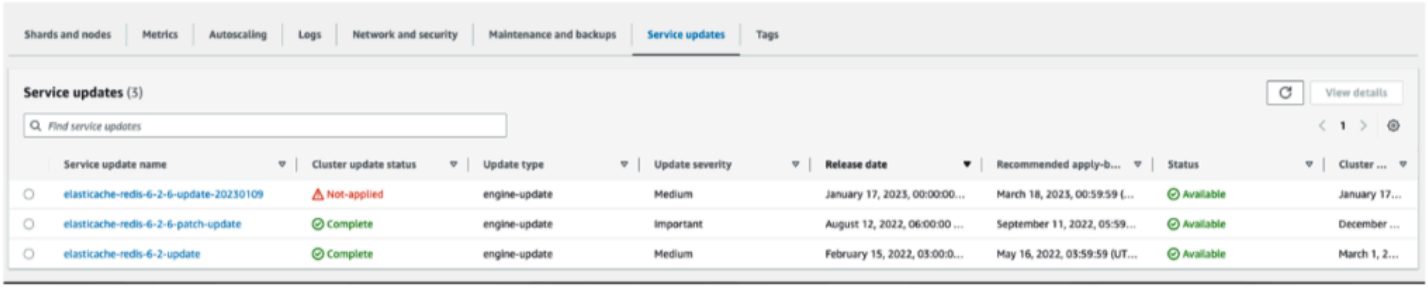
  Para obtener más información, consulte batch-apply-update-action.

Verificación de que se ha aplicado la última actualización de servicio mediante la consola de AWS

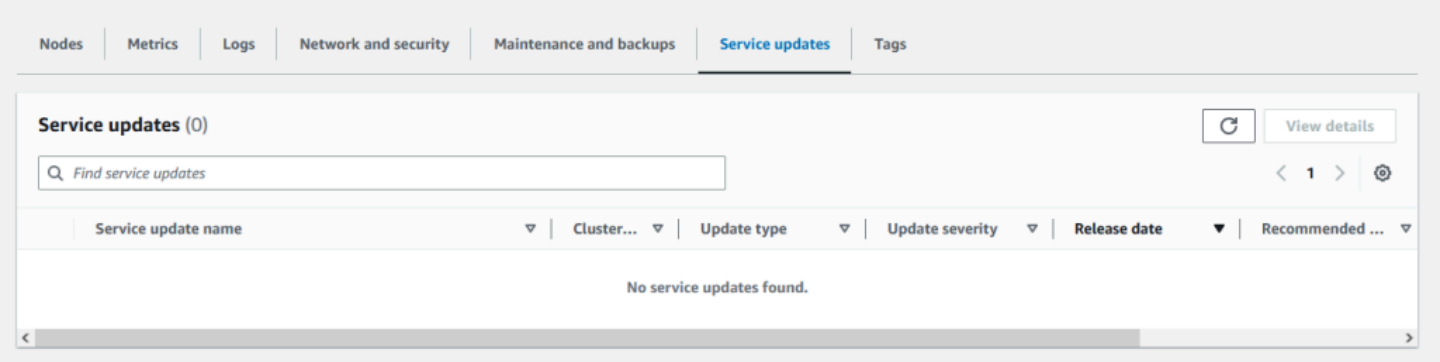
Para verificar que los clústeres de ElastiCache para Redis OSS estén ejecutando la actualización de servicio más reciente, siga estos pasos:

1. Elija un clúster aplicable en la página Clústeres de Redis OSS.
2. Seleccione Actualizaciones de servicio en el panel de navegación para ver las actualizaciones de servicio aplicables a ese clúster, si las hubiera.

Si la consola muestra una lista de actualizaciones de servicio, puede seleccionar la actualización de servicio y elegir Aplicar ahora.



Si la consola muestra el mensaje “No se ha encontrado ninguna actualización de servicio”, eso significa que el clúster de ElastiCache para Redis OSS ya tiene aplicada la última actualización de servicio.



Detención de las actualizaciones de servicio

Si es necesario, puede detener las actualizaciones de clústeres. Por ejemplo, es posible que desee detener las actualizaciones si tiene una aumento inesperado en los clústeres que se están actualizando. O es posible que desee detener las actualizaciones si tardan demasiado e interrumpen el flujo de un negocio en hora punta.

La operación Stopping (Deteniéndose) interrumpe inmediatamente todas las actualizaciones de esos clústeres y de todos los nodos que aún se deben actualizar. La operación continúa hasta completar todos los nodos que tienen el estado in progress (en curso). Sin embargo, detiene las actualizaciones de otros nodos del mismo clúster que tienen el estado update available (actualización disponible) y las cambia al estado Stopping (Deteniéndose).

Cuando se completa el flujo de trabajo Stopping (Deteniéndose), los nodos que tiene el estado Stopping (Deteniéndose) cambian al estado Stopped (Detenido). En función del flujo de trabajo de la actualización, algunos clústeres no tendrán todos los nodos actualizados. Otros clústeres podrían incluir algunos nodos que estén actualizados y otros nodos que aún tengan el estado update available (actualización disponible).

Puede terminar más adelante el proceso de actualización cuando el flujo de trabajo lo permita. En tal caso, elija los clústeres en los que desee completar las actualizaciones y, a continuación, elija Apply Now (Aplicar ahora). Para obtener más información, consulte Aplicación de las actualizaciones de servicio.

Uso de la consola

Puede interrumpir una actualización del servicio con la consola de ElastiCache. A continuación se indica cómo hacerlo:

- Después de que una actualización de servicio haya progresado en un clúster seleccionado, la consola de ElastiCache mostrará la pestaña Ver/Detener actualización, en la parte superior del panel.
- Para interrumpir la actualización, elija Stop Update (Detener actualización).
- Cuando se detenga la actualización, elija el clúster y examine el estado. Este cambia al estado Deteniéndose y al final al estado Detenido.

Mediante AWS CLI

Puede interrumpir una actualización del servicio con AWS CLI. El siguiente ejemplo de código muestra cómo hacerlo.

Para un grupo de réplica, haga lo siguiente:

```
aws elasticache batch-stop-update-action --service-update-name sample-service-update --replication-group-ids my-replication-group-1 my-replication-group-2
```

Para un clúster, haga lo siguiente:

```
aws elasticache batch-stop-update-action --service-update-name sample-service-update --cache-cluster-ids my-cache-cluster-1 my-cache-cluster-2
```

Para obtener más información, consulte BatchStopUpdateAction.

Vulnerabilidades y exposiciones comunes (CVE): vulnerabilidades de seguridad corregidas en ElastiCache

Vulnerabilidades y exposiciones comunes (CVE) es una lista de entradas de vulnerabilidades de ciberseguridad conocidas públicamente. Cada entrada es un enlace que contiene un número de identificación, una descripción y al menos una referencia pública. En esta página, encontrará una lista de vulnerabilidades de seguridad que se han corregido en ElastiCache.

Le recomendamos que actualice siempre a las versiones de ElastiCache Valkey, Redis OSS o ElastiCache Memcached más recientes para protegerse contra vulnerabilidades conocidas. Al utilizar una caché sin servidor de ElastiCache, las correcciones de CVE se aplican automáticamente a la caché. Al utilizar clústeres basados en nodos con Valkey o Redis OSS, ElastiCache expone el componente PATCH. Por ejemplo, cuando se utiliza ElastiCache para Redis OSS versión 6.2.6, 6 es la versión principal, 2 es la versión secundaria y 6 es la versión del parche. Las versiones de PATCH son para correcciones de errores compatibles con versiones anteriores y cambios no funcionales.

Puede utilizar la siguiente tabla para comprobar si una versión concreta de ElastiCache Valkey y Redis OSS tiene una corrección para una vulnerabilidad de seguridad específica. Si el clúster de ElastiCache Valkey o Redis OSS ejecuta una versión sin la corrección de seguridad, consulte la tabla siguiente y tome medidas. Puede actualizar a una versión más reciente de ElastiCache Valkey o Redis OSS que contenga la corrección o, si tiene una versión que la contiene, asegúrese de que se ha aplicado la actualización de servicio más reciente consultando Administración de actualizaciones de servicio para clústeres basados en nodos. Para obtener más información sobre las versiones del motor de ElastiCache compatibles y cómo actualizarlas, consulte Versiones del motor y actualización en ElastiCache.

> **Note**
>
> - Si un CVE se ha corregido en una versión de ElastiCache, eso significa que también se ha corregido en las versiones más recientes. Por ejemplo, si se corrige una vulnerabilidad en la versión 6.0.5 de ElastiCache para Redis OSS, esta corrección se mantiene en las versiones 6.2.6, 7.0.7 y 7.1.
> - El asterisco (*) en la siguiente tabla indica que, para solucionar la vulnerabilidad de seguridad, debe haber aplicado la actualización del servicio más reciente al clúster de ElastiCache para Redis OSS que ejecuta la versión de ElastiCache para Redis OSS especificada. Para obtener más información sobre cómo comprobar que se haya aplicado la actualización del servicio más reciente a la versión de ElastiCache para Redis OSS en

la que se ejecuta el clúster, consulte Administración de actualizaciones de servicio para clústeres basados en nodos.

Versión de ElastiCache	CVE corregidas
Valkey 8.1 y todas las versiones anteriores de Valkey Redis OSS 7.1 y todas las versiones anteriores de Redis OSS	CVE-2025-49844*, CVE-2025-46817*, CVE-2025-46818* y CVE-2025-46819*
Valkey 7.2 y 7.3	CVE-2025-21607*, CVE-2025-21605*, CVE-2024-31449*, CVE-2024-31227* y CVE-2024-31228*
Valley 7.2.7	CVE-2024-51741
Redis OSS 7.1 y 6.2	CVE-2025-21605*, CVE-2024-31449*, CVE-2024-31227*, CVE-2024-31228* y CVE-2023-41056
Redis OSS 7.0.7	CVE-2023-41056*
Redis OSS 6.2.7	CVE-2024-46981
Redis OSS 6.2.6	CVE-2022-24834*, CVE-2022-35977*, CVE-2022-36021*, CVE-2023-22458, CVE-2023-25155 y CVE-2023-28856 CVE-2023-45145: tenga en cuenta que este CVE se ha abordado en Redis OSS 6.2 y 7.0, pero no en Redis OSS 7.1.
Redis OSS 6.0.5	CVE-2022-24735*, CVE-2022-24736*

Registro y Supervisión en Amazon ElastiCache

Para administrar la memoria caché, es importante que conozca su rendimiento. ElastiCache genera métricas que se publican en Registros de Amazon CloudWatch para supervisar el rendimiento de la caché. Además, ElastiCache genera eventos cuando se producen cambios importantes en los recursos de la caché (por ejemplo, se crea o se elimina una caché).

Temas

- Métricas y eventos para cachés sin servidor de Valkey y Redis OSS
- Métricas y eventos para clústeres basados en nodos de Valkey y Redis OSS
- Eventos y métricas para cachés y clústeres de Memcached
- Registro de llamadas a la API de Amazon ElastiCache con AWS CloudTrail
- Supervisión de eventos de ElastiCache en Amazon SNS
- Entrega de registro
- Monitoreo del uso con métricas de CloudWatch

- Registro de llamadas a la API de Amazon ElastiCache con AWS CloudTrail

Métricas y eventos para cachés sin servidor de Valkey y Redis OSS

ElastiCache ofrece una amplia variedad de métricas y eventos para la supervisión al trabajar con cachés sin servidor. Esto incluye las métricas de CloudWatch, las métricas de comandos y los registros de eventos que se pueden ingerir a través de Amazon EventBridge.

Temas

- Métricas de caché sin servidor
- Eventos de caché sin servidor

Métricas de caché sin servidor

El espacio de nombres `AWS/ElastiCache` incluye las siguientes métricas de CloudWatch para las cachés sin servidor de Valkey o Redis OSS.

Códigos de métricas de Valkey o Redis OSS

Métrica	Descripción	Unidad
BytesUsedForCache	El número total de bytes utilizados por los datos almacenados en la memoria caché.	Bytes
ElastiCacheProcessingUnits	El número total de ElastiCacheProcessingUnits (ECPUs) consumidas por las solicitudes ejecutadas en la memoria caché	Recuento
SuccessfulReadRequestLatency	Latencia de las solicitudes de lectura correctas.	Microsegundos
SuccessfulWriteRequestLatency	Latencia de las solicitudes de escritura correctas	Microsegundos
TotalCmdsCount	Recuento total de todos los comandos ejecutados en la memoria caché	Recuento
CacheHitRate	Indica la tasa de aciertos de la memoria caché. Esto se calcula utilizando estadísticas `cache_hits` y `cache_misses` de la siguiente manera: `cache_hits /(cache_hits + cache_misses)` .	Porcentaje
CacheHits	El número de búsquedas de claves solo de lectura realizadas correctamente en la caché.	Recuento

Métrica	Descripción	Unidad
CurrConnections	Número de conexiones de cliente en su caché.	Recuento
ThrottledCmds	El número de solicitudes que ElastiCache limitó porque la carga de trabajo se escalaba más rápido de lo que ElastiCache podía escalar.	Recuento
NewConnections	El número total de conexione s que ha aceptado el servidor durante este periodo.	Recuento
CurrItems	El número de elementos en la caché.	Recuento
CurrVolatileItems	El número de elementos en la caché con TTL.	Recuento
NetworkBytesIn	Total de bytes transferidos a la memoria caché	Bytes
NetworkBytesOut	Total de bytes transferidos a la memoria caché	Bytes
Evictions	El recuento de claves expulsadas por la caché	Recuento
IamAuthenticationExpirations	El número total de conexione s de Valkey o Redis OSS autenticadas por IAM que han caducado. Puede encontrar más información sobre Autenticación con IAM en la guía del usuario.	Recuento

Métrica	Descripción	Unidad
IamAuthenticationThrottling	El número total de solicitudes AUTH o HELLO de Valkey o Redis OSS autenticadas por IAM con limitación. Puede encontrar más información sobre Autenticación con IAM en la guía del usuario.	Recuento
KeyAuthorizationFailures	Número total de intentos fallidos de los usuarios de acceder a claves a las que no tienen permiso para acceder. Sugerimos configurar una alarma para detectar intentos de acceso sin autorización.	Recuento
AuthenticationFailures	Número total de intentos fallidos de autenticarse en Valkey o Redis OSS mediante el comando AUTH. Sugerimos configurar una alarma para detectar intentos de acceso sin autorización.	Recuento
CommandAuthorizationFailures	Número total de intentos fallidos de los usuarios de ejecutar comandos a los que no tienen permiso para llamar. Sugerimos configurar una alarma para detectar intentos de acceso sin autorización.	Recuento

Métricas en el nivel de comando

ElastiCache también emite las siguientes métricas en el nivel de comando. Para cada tipo de comando, ElastiCache emite el recuento total de comandos y el número de ECPU consumidas por ese tipo de comando.

Métrica	Descripción	Unidad
EvalBasedCmds	El número de comandos get recibidos por la caché.	Recuento
EvalBasedCmdsECPUs	ECPU consumidas por comandos basados en eval.	Recuento
GeoSpatialBasedCmds	Número total de comandos para comandos basados en condiciones geoespaciales. Se obtiene de la estadística de commandstats de Valkey o Redis OSS. Se obtiene al sumar todos los tipos de comandos geográficos: geoadd, geodist, geohash, geopos, georadius y georadius bymember.	Recuento
GeoSpatialBasedCmdsECPUs	ECPU consumidas por comandos geoespaciales.	Recuento
GetTypeCmds	El número total de comandos de solo lectura. Se obtiene de la estadística de commandst ats de Valkey o Redis OSS mediante la suma de todos los tipos de comandos de solo lectura (get, hget, scard, lrange, etc.).	Recuento
GetTypeCmdsECPUs	Las ECPU consumidas por los comandos de lectura.	Recuento

Métrica	Descripción	Unidad
HashBasedCmds	El número total de comandos basados en hash. Se obtiene de la estadística de commandstats de Valkey o Redis OSS mediante la suma de todos los comandos que actúan en uno o más hashes (hget, hkeys, hvals, hdel, etc.).	Recuento
HashBasedCmdsECPUs	ECPU consumidas por comandos basados en hash.	Recuento
HyperLogLogBasedCmds	El número total de comandos basados en HyperLogLog. Se obtiene de la estadística de commandstats de Valkey o Redis OSS mediante la suma de todos los tipos de comandos pf (pfadd, pfcount, pfmerge, etc.).	Recuento
HyperLogLogBasedCmdsECPUs	ECPU consumidas por comandos basados en HyperLogLog.	Recuento
JsonBasedCmds	El número total de comandos JSON, incluidos los comandos de lectura y escritura. Se obtiene de la estadística de commandstats de Valkey o Redis OSS mediante la suma de todos los comandos de JSON que actúan sobre claves de JSON.	Recuento

Métrica	Descripción	Unidad
JsonBasedCmdsECPUs	ECPU consumidas por todos los comandos JSON, incluidos los comandos de lectura y escritura.	Recuento
JsonBasedGetCmds	Número total de comandos de solo lectura de JSON. Se obtiene de la estadística de commandstats de Valkey o Redis OSS mediante la suma de todos los comandos de lectura de JSON que actúan sobre claves de JSON.	Recuento
JsonBasedGetCmdsECPUs	ECPU consumidas por los comandos de solo lectura de JSON.	Recuento
JsonBasedSetCmds	Número total de comandos de escritura de JSON. Se obtiene de la estadística de commandstats de Valkey o Redis OSS mediante la suma de todos los comandos de escritura de JSON que actúan sobre claves de JSON.	Recuento
JsonBasedSetCmdsECPUs	ECPU consumidas por los comandos de escritura de JSON.	Recuento

Métrica	Descripción	Unidad
KeyBasedCmds	El número total de comandos basados en claves. Se obtiene de la estadística de commandstats de Valkey o Redis OSS mediante la suma de todos los comandos que actúan en una o más claves en varias estructuras de datos (del, expire, rename, etc.).	Recuento
KeyBasedCmdsECPUs	ECPU consumidas por comandos basados en claves.	Recuento
ListBasedCmds	El número total de comandos basados en listas. Se obtiene de la estadística de commandstats de Valkey o Redis OSS mediante la suma de todos los comandos que actúan en una o más listas (lindex, lrange, lpush, ltrim, etc.).	Recuento
ListBasedCmdsECPUs	ECPU consumidas por comandos basados en listas.	Recuento
NonKeyTypeCmds	Número total de comandos no basados en claves. Se obtiene de la estadística de commandstats de Valkey o Redis OSS mediante la suma de todos los comandos que no actúan sobre una clave, como acl, dbsize o info.	Recuento

Métrica	Descripción	Unidad
NonKeyTypeCmdsECPUs	ECPU consumidas por comandos no basados en claves.	Recuento
PubSubBasedCmds	El número total de comandos para la funcionalidad publicación/suscripción. Se obtiene de la estadística de commandstats de Valkey o Redis OSS mediante la suma de todos los comandos utilizados para la funcionalidad de publicación/suscripción: psubscribe, publish, pubsub, punsubscribe, ssubscribe, sunsubscribe, spublish, subscribe y unsubscribe.	Recuento
PubSubBasedCmdsECPUs	ECPU consumidas por comandos basados en pub/sub.	Recuento
SetBasedCmds	El número total de comandos basados en instrucciones set. Se obtiene de la estadística de commandstats de Valkey o Redis OSS mediante la suma de todos los comandos que actúan en uno o más conjuntos (scard, sdiff, sadd, sunion, etc.).	Recuento
SetBasedCmdsECPUs	ECPU consumidas por comandos basados en conjuntos.	Recuento

Métrica	Descripción	Unidad
SetTypeCmds	El número total de tipos de comandos de escritura. Se obtiene de la estadística de commandstats de Valkey o Redis OSS mediante la suma de todos los tipos de comandos mutantes que actúan en los datos (set, hset, sadd, lpop, etc.).	Recuento
SetTypeCmdsECPUs	ECPU consumidas por los comandos de escritura.	Recuento
SortedSetBasedCmds	El número total de comandos basados en instrucciones set ordenadas. Se obtiene de la estadística de commandst ats de Valkey o Redis OSS mediante la suma de todos los comandos que actúan en uno o más conjuntos ordenados (zcount, zrange, zrank, zadd, etc.).	Recuento
SortedSetBasedCmdsECPUs	ECPU consumidas por comandos basados en conjuntos ordenados.	Recuento

Métrica	Descripción	Unidad
StringBasedCmds	El número total de comandos basados en cadenas. Se obtiene de la estadística de commandstats de Valkey o Redis OSS mediante la suma de todos los comandos que actúan en una o más cadenas (strlen, setex, setrange, etc.).	Recuento
StringBasedCmdsECPUs	ECPU consumidas por comandos basados en cadenas.	Recuento
StreamBasedCmds	El número total de comandos basados en secuencias. Se obtiene de la estadística de commandstats de Valkey o Redis OSS mediante la suma de todos los comandos que actúan sobre uno o más tipos de datos de flujos (xrange, xlen, xadd, xdel, etc.).	Recuento
StreamBasedCmdsECPUs	ECPU consumidas por comandos basados en transmisiones.	Recuento

Eventos de caché sin servidor

ElastiCache registra eventos relacionados con la caché sin servidor. Esta información incluye la fecha y la hora del evento, el nombre del origen y el tipo del origen del evento, así como una descripción del evento. Puede recuperar con facilidad eventos del registro mediante la consola de ElastiCache, el comando describe-events de la AWS CLI o la acción API `DescribeEvents` de ElastiCache.

Puede elegir monitorizar, ingerir, transformar y actuar en función de los eventos de ElastiCache mediante Amazon EventBridge. Obtenga más información en la guía de introducción de Amazon EventBridge.

Visualización de eventos de ElastiCache (consola)

Para ver eventos mediante la consola de ElastiCache:

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/
2. Para ver una lista de todos los eventos disponibles, elija Events (Eventos) en el panel de navegación.
3. En la pantalla Eventos, cada fila de la lista representa un evento y muestra el origen del evento, el tipo, la hora GMT y una descripción. Con la opción Filter podrá especificar si desea ver todos los eventos o simplemente los eventos de un tipo determinado de la lista de eventos.

Visualización de eventos de ElastiCache (AWS CLI)

Para generar una lista de eventos de ElastiCache mediante la AWS CLI, utilice el comando describe-events. Puede usar parámetros opcionales para controlar el tipo de eventos que se muestran en la lista, el marco temporal de los eventos de la lista, el número máximo de eventos que se incluirán en la lista y mucho más.

El código siguiente muestra hasta 40 eventos de caché sin servidor.

```
aws elasticache describe-events --source-type serverless-cache --max-items 40
```

El código siguiente muestra todos los eventos para cachés sin servidor de las últimas 24 horas (1440 minutos).

```
aws elasticache describe-events --source-type serverless-cache --duration 1440
```

Eventos sin servidor

En esta sección, se documentan los distintos tipos de eventos que puede recibir en sus cachés sin servidor.

Eventos de creación de caché sin servidor

Tipo-detalle	Descripción	Unidad	Origen	Mensaje
Caché creada	Arn de caché	creación	caché sin servidor	La caché <cache-name> se ha creado y está lista para usarse.
Caché creada	Arn de caché Arn de la instantánea	creación	caché sin servidor	La caché <cache-name> se ha creado y los datos se han restaurado a partir de la instantánea. La caché está lista para usarse.
Error durante la creación de caché	Arn de caché	error	caché sin servidor	No se ha podido crear la memoria caché <cache-name>. No hay suficientes direcciones IP libres para crear un punto de conexión de VPC.
Error durante la creación de caché	Arn de caché	error	caché sin servidor	No se ha podido crear la memoria caché <cache-name>. Se proporcionaron subredes no válidas en la solicitud.

Tipo-detalle	Descripción	Unidad	Origen	Mensaje
Error durante la creación de caché	Arn de caché	error	caché sin servidor	No se ha podido crear la memoria caché <cache-name>. Se alcanzó el límite de cuota para crear un punto de conexión de VPC.
Error durante la creación de caché	Arn de caché	error	caché sin servidor	No se ha podido crear la memoria caché <cache-name>. No tiene permiso para crear un punto de conexión de VPC.
Error durante la creación de caché	Arn de caché	error	caché sin servidor	No se ha podido crear la memoria caché <cache-name>. En el grupo de usuarios <user-group-name>, hay un usuario con una versión de Valkey o Redis OSS incompatible.

Tipo-detalle	Descripción	Unidad	Origen	Mensaje
Error durante la creación de caché	Arn de caché Arn de instantáneas de caché	error	caché sin servidor	No se ha podido crear la memoria caché <cache-name>. El grupo de usuarios <user-group-name> proporcionado no existe.
Error durante la creación de caché	Arn de caché	error	caché sin servidor	No se ha podido crear la memoria caché <cache-name>. La restauración de los datos a partir de la instantánea falló debido a <reason>. Motivos de los errores: • error al recuperar el archivo de S3. • el md5 esperado no coincide con el md5 real. • el archivo RDB proporcionado tiene una versión no compatible.

Eventos de actualización de caché sin servidor (Valkey o Redis OSS)

Tipo-detalle	Lista de recursos	Categoría	Origen	Mensaje
Caché actualizada	Arn de caché	cambio de configuración	caché sin servidor	SecurityGroups se actualizó para la caché <cache-name>.
Caché actualizada	Arn de caché	cambio de configuración	caché sin servidor	Etiquetas actualizadas para la caché <cache-name>.
Error durante la actualización de caché	Arn de caché	cambio de configuración	caché sin servidor	Error en la actualización de la caché <cache-name>. En el grupo de usuarios <user-group-name>, hay un usuario con una versión de Valkey o Redis OSS incompatible.
Error durante la actualización de caché	Arn de caché	cambio de configuración	caché sin servidor	Error en la actualización de la caché <cache-name>. Error en la actualización de SecurityGroups.
Error durante la actualización de caché	Arn de caché	cambio de configuración	caché sin servidor	Error en la actualización de la caché <cache-

Tipo-detalle	Lista de recursos	Categoría	Origen	Mensaje
				name>. Error en la actualización de SecurityG roups por no contar con los permisos necesarios.
Error durante la actualización de caché	Arn de caché	cambio de configuración	caché sin servidor	Error en la actualización de la caché <cache-na me>. Error en la actualización de SecurityGroups debido a que los SecurityGroups no son válidos.

Eventos de eliminación de caché sin servidor (Valkey o Redis OSS)

Tipo-detalle	Lista de recursos	Categoría	Origen	Mensaje
Caché eliminada	Arn de caché	eliminación	caché sin servidor	Se ha eliminado la caché <cache-name>.

Eventos de límite de uso de caché sin servidor (Valkey o Redis OSS)

Tipo-detalle	Descripción	Unidad	Origen	Mensaje
Caché actualiza da	Arn de caché	cambio de configuración	caché sin servidor	Límites actualiza dos para la

Tipo-detalle	Descripción	Unidad	Origen	Mensaje
				caché <cache-name>.
Se acerca el límite de caché	Arn de caché	notificación	caché sin servidor	El slot <X> utiliza más del <Y> % del límite por slot de 32 GB. Por ejemplo, el slot 10 utiliza más del 90 % del límite por slot de 32 GB.
Error durante la actualización de caché	Arn de caché	error	caché sin servidor	Error en la actualización del límite en la caché <cache-name>, ya que la caché ha sido eliminada.
Error durante la actualización de caché	Arn de caché	error	caché sin servidor	Error en la actualización del límite en la caché <cache-name>, debido a una configuración no válida.

Tipo-detalle	Descripción	Unidad	Origen	Mensaje
Error durante la actualización de caché	Arn de caché	error	caché sin servidor	No se pudo actualizar el límite de la caché <cache-name> porque los datos actualmente almacenados en caché superan los nuevos límites. Vacíe algunos datos antes de aplicar los límites.

Eventos de instantáneas de caché sin servidor (Valkey o Redis OSS)

Tipo-detalle	Lista-recursos	Categoría	Origen	Mensaje
Instantánea creada	Arn de caché Arn de la instantánea	creación	serverless-cache-snapshot	Instantánea <snapshot-name> creada para la caché <cache-name>.
Error en la creación de la instantánea	Arn de caché Arn de la instantánea	error	serverless-cache-snapshot	No se ha podido crear la instantánea para la caché <cache-name>. No se ha podido crear la instantánea <snapshot-name> con la clave <key-id>

Tipo-detalle	Lista-recursos	Categoría	Origen	Mensaje
				administrada por el cliente <reason>. Mensajes sobre el motivo del error: • porque la clave administrada por el cliente está desactivada • porque no se ha podido encontrar la clave administrada por el cliente • porque se ha agotado el tiempo de espera de la solicitud

Tipo-detalle	Lista-recursos	Categoría	Origen	Mensaje
Error en la creación de la instantánea	Arn de caché Arn de la instantánea	error	serverless-cache-snapshot	No se ha podido crear la instantánea para la caché <cache-name>. La creación de la instantánea <snapshot-name> ha fallado debido a <reason>. Motivo predeterminado: • debido a un error interno
Error en la exportación de instantánea	Arn de la instantánea	error	serverless-cache-snapshot	No se ha podido exportar la instantánea para la caché <cache-name>. No se ha podido exportar la instantánea al bucket %s porque ElastiCache no tiene permisos para acceder al bucket.

Tipo-detalle	Lista-recursos	Categoría	Origen	Mensaje
Error en la exportación de instantánea	Arn de la instantánea	error	serverless-cache-snapshot	No se ha podido exportar la instantánea para la caché <cache-name>. No se ha podido exportar la instantánea al bucket '%s' porque ya hay un objeto con el mismo nombre en el bucket.
Error en la exportación de instantánea	Arn de la instantánea	error	serverless-cache-snapshot	No se ha podido exportar la instantánea para la caché <cache-name>. No se ha podido exportar la instantán ea al bucket '%s' porque el identificador de la cuenta del propietario del bucket ha cambiado.

Tipo-detalle	Lista-recursos	Categoría	Origen	Mensaje
Error en la exportación de instantánea	Arn de la instantánea	error	serverless-cache-snapshot	No se ha podido exportar la instantánea para la caché <cache-name>. No se ha podido exportar la instantánea al bucket '%s' porque no se puede acceder al bucket de S3.
Error en la exportación de instantánea	Arn de la instantánea	error	serverless-cache-snapshot	No se ha podido exportar la instantánea para la caché <cache-name>. No se ha podido exportar la instantánea al bucket '%s' porque no se puede acceder al bucket.
Error en la exportación de instantánea	Arn de la instantánea	error	serverless-cache-snapshot	No se ha podido exportar la instantánea para la caché <cache-name>. No se ha podido exportar la instantánea al bucket '%s' porque el bucket no existe.

Tipo-detalle	Lista-recursos	Categoría	Origen	Mensaje
Error en la exportación de instantánea	Arn de la instantánea	error	serverless-cache-snapshot	No se ha podido exportar la instantánea para la caché <cache-name>. No se ha podido exportar la instantánea al bucket '%s' con la clave administrada por el cliente %s de instantánea de origen <reason>.
Error en la exportación de instantánea	Arn de la instantánea	error	serverless-cache-snapshot	No se ha podido exportar la instantánea para la caché <cache-name>. No se ha podido exportar la instantánea al bucket '%s'.

Tipo-detalle	Lista-recursos	Categoría	Origen	Mensaje
Error en la copia de instantánea	Arn 1 de la instantánea Arn 2 de la instantánea	error	serverless-cache-snapshot	Error al copiar la instantán ea <snapshot-name>. No se ha podido copiar la instantánea '%s' a la instantán ea '%s' con la clave administr ada por el cliente <key-id> de la instantánea de origen <reason-n ame>.
Error en la copia de instantánea	Arn 1 de la instantánea Arn 2 de la instantánea	error	serverless-cache-snapshot	Error al copiar la instantán ea <snapshot-name>. No se ha podido copiar la instantánea '%s' a la instantán ea '%s' con la clave administr ada por el cliente '%s' '%s' de la instantánea de destino.

Métricas y eventos para clústeres basados en nodos de Valkey y Redis OSS

ElastiCache ofrece una amplia variedad de métricas y eventos para supervisar los clústeres basados en nodos de Valkey y Redis OSS. Esto incluye métricas de host, métricas de comandos y registros de eventos disponibles a través de AWS CLI y Amazon Simple Notification Service (SNS).

Temas

- Métricas para clústeres basados en nodos
- Eventos para clústeres basados en nodos (Valkey y Redis OSS)

Métricas para clústeres basados en nodos

Al crear un clúster basado en nodos, ElastiCache emite métricas en cada nivel de nodo, como las métricas de host y las métricas de caché.

Para obtener más información sobre las métricas en el nivel de host, consulte Métricas de nivel de host.

Para obtener más información sobre las métricas en el nivel de nodo, consulte Métricas de Valkey y Redis OSS.

Eventos para clústeres basados en nodos (Valkey y Redis OSS)

ElastiCache registra eventos relacionados con clústeres basados en nodos. Al trabajar con clústeres basados en nodos, puede ver los eventos del clúster en la consola de ElastiCache mediante la AWS CLI o mediante el Amazon Simple Notification Service (SNS). Los eventos de clústeres basados en nodos no se publican en Amazon EventBridge.

La información de los eventos de clústeres basados en nodos incluye la fecha y la hora del evento, el nombre y el tipo del origen del evento, así como una descripción del mismo. Puede recuperar con facilidad eventos del registro mediante la consola de ElastiCache, el comando describe-events de la AWS CLI o la acción API DescribeEvents de ElastiCache.

Visualización de eventos de ElastiCache (Consola)

El siguiente procedimiento muestra eventos mediante la consola de ElastiCache.

Para ver eventos mediante la consola de ElastiCache

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/
2. Para ver una lista de todos los eventos disponibles, elija Events (Eventos) en el panel de navegación.
3. En la pantalla Eventos, cada fila de la lista representa un evento y muestra el origen del evento, el tipo, la hora GMT y una descripción. Con la opción Filter podrá especificar si desea ver todos los eventos o simplemente los eventos de un tipo determinado de la lista de eventos.

Visualización de eventos de ElastiCache (AWS CLI)

Para generar una lista de eventos de ElastiCache mediante la AWS CLI, utilice el comando describe-events. Puede usar parámetros opcionales para controlar el tipo de eventos que se muestran en la lista, el marco temporal de los eventos de la lista, el número máximo de eventos que se incluirán en la lista y mucho más.

El código siguiente muestra hasta 40 eventos de clústeres basados en nodos.

```
aws elasticache describe-events --source-type cache-cluster --max-items 40
```

El código siguiente muestra todos los eventos para cachés basados en nodos de las últimas 24 horas (1440 minutos).

```
aws elasticache describe-events --source-type cache-cluster --duration 1440
```

Eventos de clústeres basados en nodos

Esta sección contiene la lista de eventos que podría recibir para sus clústeres basados en nodos.

Los siguientes eventos de ElastiCache desencadenan notificaciones de Amazon SNS. Para obtener más información sobre los eventos, consulte Visualización de eventos de ElastiCache.

Nombre del evento	Mensaje	Descripción
ElastiCache:AddCacheNodeComplete	`ElastiCache:AddCacheNodeComplete : cache-cluster`	Se ha agregado un nodo de caché al clúster y ya se puede usar.

Nombre del evento	Mensaje	Descripción
ElastiCache:AddCacheNodeFailed debido a direcciones IP libres insuficientes	`ElastiCache:AddCacheNodeFailed : cluster-name`	No se pudo agregar un nodo de caché porque no hay suficientes direcciones IP disponibles.
ElastiCache:CacheClusterParametersChanged	`ElastiCache:CacheClusterParametersChanged : cluster-name`	Se han cambiado uno o varios parámetros del clúster.
ElastiCache:CacheClusterProvisioningComplete	`ElastiCache:CacheClusterProvisioningComplete cluster-name-0001-005`	El aprovisionamiento de un clúster se ha completado y los nodos de caché del clúster de caché están listos para su uso.
ElastiCache:CacheClusterProvisioningFailed debido a un estado de la red incompatible	`ElastiCache:CacheClusterProvisioningFailed : cluster-name`	Se ha intentado lanzar un nuevo clúster en una nube privada virtual (VPC) que no existe.
ElastiCache:CacheClusterScalingComplete	`CacheClusterScalingComplete : cluster-name`	El escalado del clúster de caché se ha completado con éxito.
ElastiCache:CacheClusterScalingFailed	ElastiCache:CacheClusterScalingFailed : *cluster-name*	Se ha producido un error en la operación de ampliación del clúster de caché.

Nombre del evento	Mensaje	Descripción
ElastiCache:CacheClusterSecurityGroupModified	`ElastiCache:CacheClusterSecurityGroupModified : cluster-name`	Se ha producido uno de los eventos siguientes: • La lista de los grupos de seguridad de caché autorizados para el clúster se ha modificado. • Se han autorizado uno o varios de los nuevos grupos de seguridad de EC2 en alguno de los grupos de seguridad de caché asociados al clúster. • Se han rechazado uno o varios de los nuevos grupos de seguridad de EC2 de alguno de los grupos de seguridad de caché asociados al clúster.

Nombre del evento	Mensaje	Descripción
ElastiCache:CacheNodeReplaceStarted	`ElastiCache:CacheNodeReplaceStarted : cluster-name`	ElastiCache detectó que el host que ejecuta un nodo de caché tiene un rendimiento reducido o no se encuentra disponible y comenzó el reemplazo del nodo de caché. **Note** La entrada de DNS del nodo de caché de reemplazo no se ha cambiado. En la mayoría de los casos, cuando se produce este evento, no es necesario actualizar la lista de servidores de sus clientes. Sin embargo, es posible que determinadas bibliotecas de cliente de caché dejen de utilizar el nodo de caché incluso después de que ElastiCache haya reemplazado el nodo de caché. En este caso, la aplicación deberá actualizar la lista de servidores cuando se produzca este evento.

Nombre del evento	Mensaje	Descripción
ElastiCache:CacheNodeReplaceComplete	`ElastiCache:CacheNodeReplaceComplete : cluster-name`	ElastiCache detectó que el host que ejecuta un nodo de caché tiene un desempeño reducido o no se encuentra disponible y completó el reemplazo del nodo de caché. **Note** La entrada de DNS del nodo de caché de reemplazo no se ha cambiado. En la mayoría de los casos, cuando se produce este evento, no es necesario actualizar la lista de servidores de sus clientes. Sin embargo, es posible que determinadas bibliotecas de cliente de caché dejen de utilizar el nodo de caché incluso después de que ElastiCache haya reemplaza do el nodo de caché. En este caso, la aplicación deberá actualizar la lista de servidore s cuando se produzca este evento.

Nombre del evento	Mensaje	Descripción
ElastiCache:CacheNodesRebooted	`ElastiCache:CacheNodesRebooted : cluster-name`	Uno o varios nodos de caché se han reiniciado. Aparece el mensaje (Memcached): `"Cache node %s shutdown"` A continuación, aparece un segundo mensaje: `"Cache node %s restarted"`
ElastiCache:CertificateRenewalComplete (solo Valkey o Redis OSS)	`ElastiCache:CertificateRenewalComplete`	El certificado de entidad de certificación de Amazon se renovó con éxito.
ElastiCache:CreateReplicationGroupComplete	`ElastiCache:CreateReplicationGroupComplete : cluster-name`	El grupo de reproducción se ha creado correctamente.
ElastiCache:DeleteCacheClusterComplete	`ElastiCache:DeleteCacheClusterComplete : cluster-name`	Se ha completado la eliminación de un clúster y de todos los nodos de caché asociados.
ElastiCache:FailoverComplete (solo Valkey o Redis OSS)	`ElastiCache:FailoverComplete : mycluster`	La conmutación por error a un nodo de réplica se ha realizado correctamente.
ElastiCache:ReplicationGroupIncreaseReplicaCountFinished	`ElastiCache:ReplicationGroupIncreaseReplicaCountFinished : cluster-name-0001-005`	Aumentó el número de réplicas en el clúster.

Nombre del evento	Mensaje	Descripción
ElastiCache:ReplicationGroupIncreaseReplicaCountStarted	`ElastiCache:ReplicationGroupIncreaseReplicaCountStarted : cluster-name-0003-004`	Ha comenzado el proceso de agregar réplicas al clúster.
ElastiCache:NodeReplacementCanceled	`ElastiCache:NodeReplacementCanceled : cluster-name`	Un nodo del clúster que cuyo reemplazo estaba programado ya no está programado para el reemplazo.
ElastiCache:NodeReplacementRescheduled	`ElastiCache:NodeReplacementRescheduled : cluster-name`	Un nodo de su clúster que estaba programado para el reemplazo se ha vuelto a programar para el reemplazo durante el nuevo periodo descrito en la notificación. Para obtener información acerca de las acciones que puede emprender, consulte Reemplazo de nodos (Valkey y Redis OSS).
ElastiCache:NodeReplacementScheduled	`ElastiCache:NodeReplacementScheduled : cluster-name`	Un nodo de su clúster se ha programado para el reemplazo durante el periodo que se describe en la notificación. Para obtener información acerca de las acciones que puede emprender, consulte Reemplazo de nodos (Valkey y Redis OSS).

Nombre del evento	Mensaje	Descripción
ElastiCache:RemoveCacheNodeComplete	`ElastiCache:RemoveCacheNodeComplete : cluster-name`	Se ha eliminado un nodo de caché del clúster.
ElastiCache:ReplicationGroupScalingComplete	`ElastiCache:ReplicationGroupScalingComplete : cluster-name`	La operación de ampliación del grupo de reproducción se ha completado correctamente.
ElastiCache:ReplicationGroupScalingFailed	`"Failed applying modification to cache node type to %s."`	Se ha producido un error en la operación de ampliación del grupo de reproducción.
ElastiCache:ServiceUpdateAvailableForNode	`"Service update is available for cache node %s."`	Hay una actualización de autoservicio disponible para el nodo.
ElastiCache:SnapshotComplete (solo Valkey o Redis OSS)	`ElastiCache:SnapshotComplete : cluster-name`	Una instantánea de caché se ha completado correctamente.
ElastiCache:SnapshotFailed (solo Valkey o Redis OSS)	`SnapshotFailed : cluster-name`	Se ha producido un error en la instantánea de caché. Consulte los eventos de caché del clúster para obtener más detalles acerca de la causa. Si describe la instantánea (consulte `DescribeSnapshots`), el estado será `failed`.

Eventos y métricas para cachés y clústeres de Memcached

En esta sección, se describen las métricas y los eventos que se deben supervisar al trabajar con cachés sin servidor y basadas en nodos de Memcached.

Temas

- Métricas para ElastiCache sin servidor para Memcached
- Eventos para ElastiCache sin servidor para Memcached

Métricas para ElastiCache sin servidor para Memcached

En esta sección se describen las métricas y los eventos que se deben supervisar al trabajar con ElastiCache sin servidor para Memcached.

El espacio de nombres de `AWS/ElastiCache` incluye las siguientes métricas de CloudWatch para las cachés sin servidor de Memcached.

Métrica	Descripción	Unidad
BytesUsedForCache	El número total de bytes utilizados por los datos almacenados en la memoria caché.	Bytes
ElastiCacheProcessingUnits	El número total de ElastiCacheProcessingUnits (ECPUs) consumidas por las solicitudes ejecutadas en la memoria caché	Recuento
SuccessfulReadRequestLatency	Latencia de las solicitudes de lectura correctas.	Microsegundos
SuccessfulWriteRequestLatency	Latencia de las solicitudes de escritura correctas	Microsegundos

Métrica	Descripción	Unidad
TotalCmdsCount	Recuento total de todos los comandos ejecutados en la memoria caché	Recuento
CurrConnections	Número de conexiones de cliente en su caché.	Recuento
ThrottledCmds	El número de solicitudes que ElastiCache limitó porque la carga de trabajo se escalaba más rápido de lo que ElastiCache podía escalar.	Recuento
NewConnections	El número total de conexione s que ha aceptado el servidor durante este periodo.	Recuento
CurrItems	El número de elementos en la caché.	Recuento
NetworkBytesIn	Total de bytes transferidos a la memoria caché	Bytes
NetworkBytesOut	Total de bytes transferidos a la memoria caché	Bytes
Evictions	El recuento de claves expulsadas por la caché	Recuento
Reclaimed	El número de claves declaradas como vencidas por la caché	Recuento

Métricas en el nivel de comando

ElastiCache también emite las siguientes métricas de Memcached en el nivel de comando

Métrica	Descripción	Unidad
CmdGet	El número de comandos get recibidos por la caché.	Recuento
CmdSet	El número de comandos establecidos que ha recibido la caché.	Recuento
CmdTouch	El número de comandos de consulta (touch) recibidos por la caché.	Recuento
GetHits	El número de solicitudes get que la caché ha recibido en las que se encontró la clave solicitada.	Recuento
GetMisses	El número de solicitudes get que la caché ha recibido en las que no se encontró la clave solicitada.	Recuento
IncrHits	El número de solicitudes increment que la caché ha recibido en las que se encontró la clave solicitada.	Recuento
IncrMisses	El número de solicitudes increment que la caché ha recibido en las que no se encontró la clave solicitada.	Recuento
DecrHits	El número de solicitudes decrement que la caché ha recibido en las que se encontró la clave solicitada.	Recuento

Métrica	Descripción	Unidad
DecrMisses	El número de solicitudes decrement que la caché ha recibido en las que no se encontró la clave solicitada.	Recuento
DeleteHits	El número de solicitud es delete que la caché ha recibido en las que se encontró la clave solicitada.	Recuento
DeleteMisses	El número de solicitudes delete que la caché ha recibido en las que no se encontró la clave solicitada.	Recuento
TouchHits	El número de clases que se han manipulado y que recibieron una nueva hora de vencimiento.	Recuento
TouchMisses	El número de claves consultad as, pero no encontradas.	Recuento
CasHits	El número de solicitudes cas que la caché ha recibido en las que se encontró la clave solicitada y en las que el valor de cas coincidía.	Recuento
CasMisses	El número de solicitudes cas que la caché ha recibido en las que se no encontró la clave solicitada.	Recuento

Métrica	Descripción	Unidad
CasBadval	El número de solicitudes cas que la caché ha recibido en las que el valor de cas no coincidía con el valor de cas almacenado.	Recuento
CmdFlush	El número de comandos flush recibidos por la caché.	Recuento

Eventos para ElastiCache sin servidor para Memcached

ElastiCache registra eventos relacionados con la caché sin servidor. Esta información incluye la fecha y la hora del evento, el nombre del origen y el tipo del origen del evento, así como una descripción del evento. Puede recuperar con facilidad eventos del registro mediante la consola de ElastiCache, el comando describe-events de la AWS CLI o la acción API `DescribeEvents` de ElastiCache.

Puede elegir monitorizar, ingerir, transformar y actuar en función de los eventos de ElastiCache mediante Amazon EventBridge. Obtenga más información en la guía de introducción de Amazon EventBridge.

Visualización de eventos de ElastiCache (consola)

Para ver eventos mediante la consola de ElastiCache:

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/
2. Para ver una lista de todos los eventos disponibles, elija Events (Eventos) en el panel de navegación.
3. En la pantalla Eventos, cada fila de la lista representa un evento y muestra el origen del evento, el tipo, la hora GMT y una descripción. Con la opción Filter podrá especificar si desea ver todos los eventos o simplemente los eventos de un tipo determinado de la lista de eventos.

Visualización de eventos de ElastiCache (AWS CLI)

Para generar una lista de eventos de ElastiCache mediante la AWS CLI, utilice el comando describe-events. Puede usar parámetros opcionales para controlar el tipo de eventos que se muestran en la lista, el marco temporal de los eventos de la lista, el número máximo de eventos que se incluirán en la lista y mucho más.

El código siguiente muestra hasta 40 eventos de caché sin servidor.

```
aws elasticache describe-events --source-type serverless-cache --max-items 40
```

El código siguiente muestra todos los eventos para cachés sin servidor de las últimas 24 horas (1440 minutos).

```
aws elasticache describe-events --source-type serverless-cache --duration 1440
```

Eventos sin servidor

En esta sección, se documentan los distintos tipos de eventos que puede recibir en sus cachés sin servidor.

Eventos de creación de caché sin servidor

Tipo-detalle	Descripción	Unidad	Origen	Mensaje
Caché creada	Arn de caché	creación	caché sin servidor	La caché <cache-name> se ha creado y está lista para usarse.
Error durante la creación de caché	Arn de caché	error	caché sin servidor	No se ha podido crear la memoria caché <cache-name>. No hay suficient es direcciones IP libres para crear un punto de conexión de VPC.

Tipo-detalle	Descripción	Unidad	Origen	Mensaje
Error durante la creación de caché	Arn de caché	error	caché sin servidor	No se ha podido crear la memoria caché <cache-name>. Se proporcionaron subredes no válidas en la solicitud.
Error durante la creación de caché	Arn de caché	error	caché sin servidor	No se ha podido crear la memoria caché <cache-name>. Se alcanzó el límite de cuota para crear un punto de conexión de VPC.
Error durante la creación de caché	Arn de caché	error	caché sin servidor	No se ha podido crear la memoria caché <cache-name>. No tiene permiso para crear un punto de conexión de VPC.

Eventos de actualización de caché sin servidor (Memcached)

Tipo-detalle	Lista de recursos	Categoría	Origen	Mensaje
Caché actualizada	Arn de caché	cambio de configuración	caché sin servidor	SecurityGroups se actualizó para

Tipo-detalle	Lista de recursos	Categoría	Origen	Mensaje
				la caché <cache-name>.
Caché actualizada	Arn de caché	cambio de configuración	caché sin servidor	Etiquetas actualizadas para la caché <cache-name>.
Error durante la actualización de caché	Arn de caché	cambio de configuración	caché sin servidor	Error en la actualización de la caché <cache-name>. Error en la actualización de SecurityGroups.
Error durante la actualización de caché	Arn de caché	cambio de configuración	caché sin servidor	Error en la actualización de la caché <cache-name>. Error en la actualización de SecurityGroups por no contar con los permisos necesarios.

Tipo-detalle	Lista de recursos	Categoría	Origen	Mensaje
Error durante la actualización de caché	Arn de caché	cambio de configuración	caché sin servidor	Error en la actualización de la caché <cache-name>. Error en la actualización de SecurityGroups debido a que los SecurityGroups no son válidos.

Eventos de eliminación de caché sin servidor (Memcached)

Tipo-detalle	Lista de recursos	Categoría	Origen	Mensaje
Caché eliminada	Arn de caché	eliminación	caché sin servidor	Se ha eliminado la caché <cache-name>.

Eventos de límite de uso de caché sin servidor (Memcached)

Tipo-detalle	Descripción	Unidad	Origen	Mensaje
Caché actualizada	Arn de caché	cambio de configuración	caché sin servidor	Límites actualizados para la caché <cache-name>.
Error durante la actualización de caché	Arn de caché	error	caché sin servidor	Error en la actualización del límite en la caché <cache-name>, ya que

Tipo-detalle	Descripción	Unidad	Origen	Mensaje
				la caché ha sido eliminada.
Error durante la actualización de caché	Arn de caché	error	caché sin servidor	Error en la actualización del límite en la caché <cache-name>, debido a una configuración no válida.

Eventos de instantáneas de caché sin servidor (Memcached)

Tipo-detalle	Lista-recursos	Categoría	Origen	Mensaje
Instantánea creada	Arn de caché Arn de la instantánea	creación	serverless-cache-snapshot	Instantánea <snapshot-name> creada para la caché <cache-name>.
Error en la creación de la instantánea	Arn de caché Arn de la instantánea	error	serverless-cache-snapshot	No se ha podido crear la instantánea para la caché <cache-name>. No se ha podido crear la instantánea <snapshot-name> con la clave <key-id> administrada por el cliente <reason>.

Tipo-detalle	Lista-recursos	Categoría	Origen	Mensaje
				Mensajes sobre el motivo del error: • porque la clave administrada por el cliente está desactivada • porque no se ha podido encontrar la clave administrada por el cliente • porque se ha agotado el tiempo de espera de la solicitud

Tipo-detalle	Lista-recursos	Categoría	Origen	Mensaje
Error en la creación de la instantánea	Arn de caché Arn de la instantánea	error	serverless-cache-snapshot	No se ha podido crear la instantánea para la caché <cache-name>. La creación de la instantánea <snapshot-name> ha fallado debido a <reason>. Motivo predeterminado: • debido a un error interno
Error en la exportación de instantánea	Arn de la instantánea	error	serverless-cache-snapshot	No se ha podido exportar la instantánea para la caché <cache-name>. No se ha podido exportar la instantánea al bucket %s porque ElastiCache no tiene permisos para acceder al bucket.

Tipo-detalle	Lista-recursos	Categoría	Origen	Mensaje
Error en la exportación de instantánea	Arn de la instantánea	error	serverless-cache-snapshot	No se ha podido exportar la instantánea para la caché <cache-name>. No se ha podido exportar la instantánea al bucket '%s' porque ya hay un objeto con el mismo nombre en el bucket.
Error en la exportación de instantánea	Arn de la instantánea	error	serverless-cache-snapshot	No se ha podido exportar la instantánea para la caché <cache-name>. No se ha podido exportar la instantán ea al bucket '%s' porque el identificador de la cuenta del propietario del bucket ha cambiado.

Tipo-detalle	Lista-recursos	Categoría	Origen	Mensaje
Error en la exportación de instantánea	Arn de la instantánea	error	serverless-cache-snapshot	No se ha podido exportar la instantánea para la caché <cache-name>. No se ha podido exportar la instantánea al bucket '%s' porque no se puede acceder al bucket de S3.
Error en la exportación de instantánea	Arn de la instantánea	error	serverless-cache-snapshot	No se ha podido exportar la instantánea para la caché <cache-name>. No se ha podido exportar la instantánea al bucket '%s' porque no se puede acceder al bucket.
Error en la exportación de instantánea	Arn de la instantánea	error	serverless-cache-snapshot	No se ha podido exportar la instantánea para la caché <cache-name>. No se ha podido exportar la instantánea al bucket '%s' porque el bucket no existe.

Tipo-detalle	Lista-recursos	Categoría	Origen	Mensaje
Error en la exportación de instantánea	Arn de la instantánea	error	serverless-cache-snapshot	No se ha podido exportar la instantánea para la caché <cache-name>. No se ha podido exportar la instantánea al bucket '%s' con la clave administrada por el cliente %s de instantánea de origen <reason>.
Error en la exportación de instantánea	Arn de la instantánea	error	serverless-cache-snapshot	No se ha podido exportar la instantánea para la caché <cache-name>. No se ha podido exportar la instantánea al bucket '%s'.

Tipo-detalle	Lista-recursos	Categoría	Origen	Mensaje
Error en la copia de instantánea	Arn 1 de la instantánea Arn 2 de la instantánea	error	serverless-cache-snapshot	Error al copiar la instantán ea <snapshot-name>. No se ha podido copiar la instantánea '%s' a la instantán ea '%s' con la clave administr ada por el cliente <key-id> de la instantánea de origen <reason-n ame>.
Error en la copia de instantánea	Arn 1 de la instantánea Arn 2 de la instantánea	error	serverless-cache-snapshot	Error al copiar la instantán ea <snapshot-name>. No se ha podido copiar la instantánea '%s' a la instantán ea '%s' con la clave administr ada por el cliente '%s' '%s' de la instantánea de destino.

Registro de llamadas a la API de Amazon ElastiCache con AWS CloudTrail

Amazon ElastiCache se integra a AWS CloudTrail, un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un servicio de AWS en Amazon ElastiCache. CloudTrail

obtiene todas las llamadas a la API para Amazon ElastiCache como eventos, incluido las llamadas procedentes de la consola de Amazon ElastiCache y de las llamadas de código a las operaciones de la API de Amazon ElastiCache. Si crea un registro de seguimiento, puede habilitar la entrega continua de eventos de CloudTrail a un bucket de Amazon S3, incluido los eventos de Amazon ElastiCache. Si no configura un registro de seguimiento, puede ver los eventos más recientes de la consola de CloudTrail en el Historial de eventos. Mediante la información recopilada por CloudTrail, puede determinar la solicitud que se realizó a Amazon ElastiCache, la dirección IP de origen desde la que se realizó, quién la realizó y cuándo se realizó y otros detalles adicionales.

Para obtener más información sobre CloudTrail, consulte la Guía del usuario de AWS CloudTrail.

Información sobre Amazon ElastiCache en CloudTrail

CloudTrail se habilita en su cuenta de AWS cuando la crea. Cuando se produce una activación en Amazon ElastiCache, se registra en un evento de CloudTrail junto con los eventos de los demás servicios de AWS en Event history (Historial de eventos). Puede ver, buscar y descargar los últimos eventos de la cuenta de AWS. Para obtener más información, consulte Visualización de eventos con el historial de eventos de CloudTrail.

Para mantener un registro continuo de los eventos de la cuenta de AWS, incluido los eventos de Amazon ElastiCache, cree un registro de seguimiento. Un registro de seguimiento permite a CloudTrail enviar archivos de registro a un bucket de Amazon S3. De forma predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las regiones. El registro de seguimiento registra los eventos de todas las regiones de la partición de AWS y envía los archivos de registro al bucket de Amazon S3 especificado. También es posible configurar otros servicios de AWS para analizar en profundidad y actuar en función de los datos de eventos recopilados en los registros de CloudTrail. Para más información, consulte los siguientes temas:

- Introducción a la creación de registros de seguimiento
- Servicios e integraciones compatibles con CloudTrail
- Configuración de notificaciones de Amazon SNS para CloudTrail
- Recepción de archivos de registro de CloudTrail de varias regiones y Recepción de archivos de registro de CloudTrail de varias cuentas

CloudTrail registra todas las acciones de Amazon ElastiCache y se encuentran documentadas en la Referencia de la API de ElastiCache. Por ejemplo, las llamadas a las acciones

`CreateCacheCluster`, `DescribeCacheCluster` y `ModifyCacheCluster` generan entradas en los archivos de registros de CloudTrail.

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario lo ayuda a determinar lo siguiente:

- Si la solicitud se realizó con las credenciales raíz o del usuario de IAM.
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro servicio de AWS.

Para obtener más información, consulte el Elemento userIdentity de CloudTrail.

Descripción de las entradas de archivo de registros de Amazon ElastiCache

Un registro de seguimiento es una configuración que permite la entrega de eventos como archivos de registros en un bucket de Amazon S3 que especifique. Los archivos de registro de CloudTrail pueden contener una o varias entradas de registro. Un evento representa una solicitud específica realizada desde un origen y contiene información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etc. Los archivos de registro de CloudTrail no son un rastro de pila ordenado de las llamadas a las API públicas, por lo que no aparecen en ningún orden específico.

En el ejemplo siguiente, se muestra una entrada de registro de CloudTrail que ilustra la acción `CreateCacheCluster`.

```
{
    "eventVersion":"1.01",
    "userIdentity":{
        "type":"IAMUser",
        "principalId":"EXAMPLEEXAMPLEEXAMPLE",
        "arn":"arn:aws:iam::123456789012:user/elasticache-allow",
        "accountId":"123456789012",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "userName":"elasticache-allow"
    },
    "eventTime":"2014-12-01T22:00:35Z",
    "eventSource":"elasticache.amazonaws.com",
    "eventName":"CreateCacheCluster",
    "awsRegion":"us-west-2",
    "sourceIPAddress":"192.0.2.01",
```

```
    "userAgent":"AWS CLI/ElastiCache 1.10 API 2014-12-01",
    "requestParameters":{
        "numCacheNodes":2,
        "cacheClusterId":"test-memcached",
        "engine":"memcached",
        "aZMode":"cross-az",
        "cacheNodeType":"cache.m1.small",

    },
    "responseElements":{
        "engine":"memcached",
        "clientDownloadLandingPage":"https://console.aws.amazon.com/elasticache/
home#client-download:",
        "cacheParameterGroup":{
            "cacheParameterGroupName":"default.memcached1.4",
            "cacheNodeIdsToReboot":{
            },
            "parameterApplyStatus":"in-sync"
        },
        "preferredAvailabilityZone":"Multiple",
        "numCacheNodes":2,
        "cacheNodeType":"cache.m1.small",

        "cacheClusterStatus":"creating",
        "autoMinorVersionUpgrade":true,
        "preferredMaintenanceWindow":"thu:05:00-thu:06:00",
        "cacheClusterId":"test-memcached",
        "engineVersion":"1.4.14",
        "cacheSecurityGroups":[
            {
                "status":"active",
                "cacheSecurityGroupName":"default"
            }
        ],
        "pendingModifiedValues":{
        }
    },
    "requestID":"104f30b3-3548-11e4-b7b8-6d79ffe84edd",
    "eventID":"92762127-7a68-42ce-8787-927d2174cde1"
}
```

En el siguiente ejemplo, se muestra una entrada de registro de CloudTrail que ilustra la acción `DescribeCacheCluster`. Tenga en cuenta que para todas las llamadas de Amazon ElastiCache Describe (`Describe*`), la sección `ResponseElements` se elimina y aparece como `null`.

```
{
    "eventVersion":"1.01",
    "userIdentity":{
        "type":"IAMUser",
        "principalId":"EXAMPLEEXAMPLEEXAMPLE",
        "arn":"arn:aws:iam::123456789012:user/elasticache-allow",
        "accountId":"123456789012",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "userName":"elasticache-allow"
    },
    "eventTime":"2014-12-01T22:01:00Z",
    "eventSource":"elasticache.amazonaws.com",
    "eventName":"DescribeCacheClusters",
    "awsRegion":"us-west-2",
    "sourceIPAddress":"192.0.2.01",
    "userAgent":"AWS CLI/ElastiCache 1.10 API 2014-12-01",
    "requestParameters":{
        "showCacheNodeInfo":false,
        "maxRecords":100
    },
    "responseElements":null,
    "requestID":"1f0b5031-3548-11e4-9376-c1d979ba565a",
    "eventID":"a58572a8-e81b-4100-8e00-1797ed19d172"
}
```

En el ejemplo siguiente se muestra una entrada de log de CloudTrail que registra una acción `ModifyCacheCluster`.

```
{
    "eventVersion":"1.01",
    "userIdentity":{
        "type":"IAMUser",
        "principalId":"EXAMPLEEXAMPLEEXAMPLE",
        "arn":"arn:aws:iam::123456789012:user/elasticache-allow",
        "accountId":"123456789012",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "userName":"elasticache-allow"
    },
```

```
    "eventTime":"2014-12-01T22:32:21Z",
    "eventSource":"elasticache.amazonaws.com",
    "eventName":"ModifyCacheCluster",
    "awsRegion":"us-west-2",
    "sourceIPAddress":"192.0.2.01",
    "userAgent":"AWS CLI/ElastiCache 1.10 API 2014-12-01",
    "requestParameters":{
        "applyImmediately":true,
        "numCacheNodes":3,
        "cacheClusterId":"test-memcached"
    },
    "responseElements":{
        "engine":"memcached",
        "clientDownloadLandingPage":"https://console.aws.amazon.com/elasticache/
home#client-download:",
        "cacheParameterGroup":{
            "cacheParameterGroupName":"default.memcached1.4",
            "cacheNodeIdsToReboot":{
            },
            "parameterApplyStatus":"in-sync"
        },
        "cacheClusterCreateTime":"Dec 1, 2014 10:16:06 PM",
        "preferredAvailabilityZone":"Multiple",
        "numCacheNodes":2,
        "cacheNodeType":"cache.m1.small",
        "cacheClusterStatus":"modifying",
        "autoMinorVersionUpgrade":true,
        "preferredMaintenanceWindow":"thu:05:00-thu:06:00",
        "cacheClusterId":"test-memcached",
        "engineVersion":"1.4.14",
        "cacheSecurityGroups":[
            {
                "status":"active",
                "cacheSecurityGroupName":"default"
            }
        ],
        "configurationEndpoint":{
            "address":"test-memcached.example.cfg.use1prod.cache.amazonaws.com",
            "port":11211
        },
        "pendingModifiedValues":{
            "numCacheNodes":3
        }
    },
```

```
    "requestID":"807f4bc3-354c-11e4-9376-c1d979ba565a",
    "eventID":"e9163565-376f-4223-96e9-9f50528da645"
}
```

Supervisión de eventos de ElastiCache en Amazon SNS

Cuando se producen eventos significativos en un clúster, ElastiCache envía una notificación a un tema de Amazon SNS concreto. Por ejemplo, errores al agregar un nodo, adiciones de nodos correctas, cambios en un grupo de seguridad, etc. Al monitorear los eventos clave, podrá conocer el estado actual de los clústeres y, dependiendo del evento, adoptar medidas correctivas.

Temas

- Administración de notificaciones de Amazon SNS de ElastiCache
- Visualización de eventos de ElastiCache
- Notificaciones de eventos y Amazon SNS

Administración de notificaciones de Amazon SNS de ElastiCache

Puede configurar ElastiCache para enviar notificaciones de los eventos de clúster importantes mediante Amazon Simple Notification Service (Amazon SNS). En estos ejemplos, podrá configurar un clúster con el nombre de recurso de Amazon (ARN) de un tema de Amazon SNS para recibir notificaciones.

> **Note**
>
> - En este tema, se da por sentado que se registró en Amazon SNS, que configuró un tema de Amazon SNS y se suscribió a dicho tema. Para obtener más información sobre cómo realizar esto, consulte la Guía para desarrolladores de Amazon Simple Notification
> - De forma predeterminada, `API modify-replication-group` afecta a todos los grupos de una región y no solo al grupo especificado actualmente. Si desea configurar un solo grupo específico de una región de forma diferente a los demás grupos, puede utilizar la opción `--notification-topic-arn` para crear un tema independiente para ese grupo.

Adición de un tema de Amazon SNS

En las siguientes secciones se muestra cómo agregar un tema de Amazon SNS mediante la consola de AWS, la AWS CLI o la API de ElastiCache.

Adición de un tema de Amazon SNS (Consola)

En el siguiente procedimiento se muestra cómo agregar un tema de Amazon SNS para un clúster. Cuando utilice Valkey o Redis OSS para añadir un tema de Amazon SNS para un grupo de replicación en el paso 2, en lugar de elegir un clúster, elija un grupo de replicación. A continuación, siga los mismos pasos restantes.

> **Note**
>
> Este proceso también se puede utilizar para modificar el tema de Amazon SNS.

A fin de agregar o modificar un tema de Amazon SNS para un clúster (Consola)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En Clusters (Clústeres), elija el clúster en el que desee agregar o modificar un ARN de tema de Amazon SNS.
3. Elija Modificar.
4. En Modify Cluster (Modificar clúster) en Topic for SNS Notification (Tema para notificación SNS), elija el tema de SNS que desea agregar, o bien elija Manual ARN input (Entrada manual de ARN) y escriba el ARN del tema de Amazon SNS.
5. Elija Modificar.

Adición de un tema de Amazon SNS (AWS CLI)

A fin de agregar o modificar un tema de Amazon SNS para un clúster, utilice el comando de la AWS CLI `modify-cache-cluster`.

El siguiente ejemplo de código agrega un ARN de tema de Amazon SNS a my-cluster.

Para Linux, macOS o Unix:

```
aws elasticache modify-cache-cluster \
```

```
    --cache-cluster-id my-cluster \
    --notification-topic-arn arn:aws:sns:us-
west-2:123456789xxx:ElastiCacheNotifications
```

Para Windows:

```
aws elasticache modify-cache-cluster ^
    --cache-cluster-id my-cluster ^
    --notification-topic-arn arn:aws:sns:us-west-2:123456789xx:ElastiCacheNotifications
```

Para obtener más información, consulte modify-cache-cluster.

Adición de un tema de Amazon SNS (API de ElastiCache)

A fin de agregar o modificar un tema de Amazon SNS para un clúster, realice una llamada a la acción `ModifyCacheCluster` con los siguientes parámetros:

- `CacheClusterId=my-cluster`
- `TopicArn=arn%3Aaws%3Asns%3Aus-west-2%3A565419523791%3AElastiCacheNotifications`

Example

```
https://elasticache.amazon.com/
    ?Action=ModifyCacheCluster
    &ApplyImmediately=false
    &CacheClusterId=my-cluster
    &NotificationTopicArn=arn%3Aaws%3Asns%3Aus-
west-2%3A565419523791%3AElastiCacheNotifications
    &Version=2014-12-01
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20141201T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Para obtener más información, consulte ModifyCacheCluster.

Habilitación y deshabilitación de notificaciones de Amazon SNS

Puede habilitar o deshabilitar las notificaciones para un clúster. Los siguientes procedimientos muestran cómo deshabilitar las notificaciones de Amazon SNS.

Habilitación y deshabilitación de las notificaciones de Amazon SNS (Consola)

Para deshabilitar las notificaciones de Amazon SNS mediante Consola de administración de AWS

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. Para ver una lista de clústeres que ejecutan Memcached, elija Memcached en el panel de navegación.

   Para ver una lista de clústeres que ejecutan Valkey o Redis OSS, elija Valkey o Redis OSS en el panel de navegación.
3. Elija la casilla situada a la izquierda del nombre del clúster cuya notificación desee modificar.
4. Elija Modificar.
5. En Modify Cluster, en Topic for SNS Notification, elija Disable Notifications.
6. Elija Modificar.

Habilitación y deshabilitación de las notificaciones de Amazon SNS (AWS CLI)

Para deshabilitar las notificaciones de Amazon SNS, utilice el comando `modify-cache-cluster` con los siguientes parámetros:

Para Linux, macOS o Unix:

```
aws elasticache modify-cache-cluster \
    --cache-cluster-id my-cluster \
    --notification-topic-status inactive
```

Para Windows:

```
aws elasticache modify-cache-cluster ^
    --cache-cluster-id my-cluster ^
    --notification-topic-status inactive
```

> **Note**
>
> Si el clúster pertenece a un grupo de replicación, debe usar el comando de la CLI `modify-replication-group` para habilitar o deshabilitar las notificaciones de SNS.

Habilitación y deshabilitación de las notificaciones de Amazon SNS (API de ElastiCache)

Para deshabilitar las notificaciones de Amazon SNS, realice una llamada a la acción `ModifyCacheCluster` con los siguientes parámetros:

- `CacheClusterId=my-cluster`
- `NotificationTopicStatus=inactive`

Esta llamada devuelve un resultado similar al siguiente:

Example

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=ModifyCacheCluster
    &ApplyImmediately=false
    &CacheClusterId=my-cluster
    &NotificationTopicStatus=inactive
    &Version=2014-12-01
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20141201T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Visualización de eventos de ElastiCache

ElastiCache registra eventos relacionados con instancias del clúster, grupos de seguridad y grupos de parámetros. Esta información incluye la fecha y la hora del evento, el nombre del origen y el tipo del origen del evento, así como una descripción del evento. Puede recuperar con facilidad eventos del registro mediante la consola de ElastiCache, el comando de la AWS CLI `describe-events` o la acción de la API de ElastiCache `DescribeEvents`.

Los siguientes procedimientos muestran cómo visualizar todos los eventos de ElastiCache de las últimas 24 horas (1440 minutos).

Visualización de eventos de ElastiCache (Consola)

El siguiente procedimiento muestra eventos mediante la consola de ElastiCache.

Para ver eventos mediante la consola de ElastiCache

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. Para ver una lista de todos los eventos disponibles, elija Events (Eventos) en el panel de navegación.

   En la pantalla Events (Eventos), cada fila de la lista representa un evento y muestra el origen del evento, el tipo de evento (cache-cluster, cache-parameter-group, cache-security-group o cache-subnet-group), la hora GMT del evento y la descripción de este último.

   Con la opción Filter podrá especificar si desea ver todos los eventos o simplemente los eventos de un tipo determinado de la lista de eventos.

Visualización de eventos de ElastiCache (AWS CLI)

Para generar una lista de eventos de ElastiCache mediante la AWS CLI, utilice el comando `describe-events`. Puede usar parámetros opcionales para controlar el tipo de eventos que se muestran en la lista, el marco temporal de los eventos de la lista, el número máximo de eventos que se incluirán en la lista y mucho más.

El código siguiente muestra hasta 40 eventos de clúster.

```
aws elasticache describe-events --source-type cache-cluster --max-items 40
```

El código siguiente muestra todos los eventos de las últimas 24 horas (1440 minutos).

```
aws elasticache describe-events --source-type cache-cluster --duration 1440
```

La salida del comando `describe-events` es similar a la siguiente.

```
aws elasticache describe-events --source-type cache-cluster --max-items 40
{
    "Events": [
        {
            "SourceIdentifier": "my-mem-cluster",
            "SourceType": "cache-cluster",
            "Message": "Finished modifying number of nodes from 1 to 3",
            "Date": "2020-06-09T02:01:21.772Z"
        },
        {
            "SourceIdentifier": "my-mem-cluster",
            "SourceType": "cache-cluster",
            "Message": "Added cache node 0002 in availability zone us-west-2a",
            "Date": "2020-06-09T02:01:21.716Z"
        },
        {
            "SourceIdentifier": "my-mem-cluster",
            "SourceType": "cache-cluster",
            "Message": "Added cache node 0003 in availability zone us-west-2a",
            "Date": "2020-06-09T02:01:21.706Z"
        },
        {
            "SourceIdentifier": "my-mem-cluster",
            "SourceType": "cache-cluster",
            "Message": "Increasing number of requested nodes",
            "Date": "2020-06-09T01:58:34.178Z"
        },
        {
            "SourceIdentifier": "mycluster-0003-004",
            "SourceType": "cache-cluster",
            "Message": "Added cache node 0001 in availability zone us-west-2c",
            "Date": "2020-06-09T01:51:14.120Z"
        },
        {
            "SourceIdentifier": "mycluster-0003-004",
            "SourceType": "cache-cluster",
```

```
            "Message": "This cluster does not support persistence (ex: 'appendonly').
 Please use a different instance type to enable persistence.",
            "Date": "2020-06-09T01:51:14.095Z"
        },
        {
            "SourceIdentifier": "mycluster-0003-004",
            "SourceType": "cache-cluster",
            "Message": "Cache cluster created",
            "Date": "2020-06-09T01:51:14.094Z"
        },
        {
            "SourceIdentifier": "mycluster-0001-005",
            "SourceType": "cache-cluster",
            "Message": "Added cache node 0001 in availability zone us-west-2b",
            "Date": "2020-06-09T01:42:55.603Z"
        },
        {
            "SourceIdentifier": "mycluster-0001-005",
            "SourceType": "cache-cluster",
            "Message": "This cluster does not support persistence (ex: 'appendonly').
 Please use a different instance type to enable persistence.",
            "Date": "2020-06-09T01:42:55.576Z"
        },
        {
            "SourceIdentifier": "mycluster-0001-005",
            "SourceType": "cache-cluster",
            "Message": "Cache cluster created",
            "Date": "2020-06-09T01:42:55.574Z"
        },
        {
            "SourceIdentifier": "mycluster-0001-004",
            "SourceType": "cache-cluster",
            "Message": "Added cache node 0001 in availability zone us-west-2b",
            "Date": "2020-06-09T01:28:40.798Z"
        },
        {
            "SourceIdentifier": "mycluster-0001-004",
            "SourceType": "cache-cluster",
            "Message": "This cluster does not support persistence (ex: 'appendonly').
 Please use a different instance type to enable persistence.",
            "Date": "2020-06-09T01:28:40.775Z"
        },
        {
            "SourceIdentifier": "mycluster-0001-004",
```

```
            "SourceType": "cache-cluster",
            "Message": "Cache cluster created",
            "Date": "2020-06-09T01:28:40.773Z"
        }
    ]
}
```

Para obtener más información como, por ejemplo, los parámetros disponibles y los valores de parámetros permitidos, consulte `describe-events`.

Visualización de eventos de ElastiCache (API de ElastiCache)

Para generar una lista de eventos de ElastiCache mediante la API de ElastiCache, utilice la acción `DescribeEvents`. Puede usar parámetros opcionales para controlar el tipo de eventos que se muestran en la lista, el marco temporal de los eventos de la lista, el número máximo de eventos que se incluirán en la lista y mucho más.

El código siguiente muestra los 40 eventos de clúster de caché más recientes.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DescribeEvents
   &MaxRecords=40
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &SourceType=cache-cluster
   &Timestamp=20150202T192317Z
   &Version=2015-02-02
   &X-Amz-Credential=<credential>
```

El código siguiente muestra los eventos de clúster de caché de las últimas 24 horas (1440 minutos).

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DescribeEvents
   &Duration=1440
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &SourceType=cache-cluster
   &Timestamp=20150202T192317Z
   &Version=2015-02-02
   &X-Amz-Credential=<credential>
```

Las acciones anteriores deberían producir una salida similar a la siguiente.

```
<DescribeEventsResponse xmlns="http://elasticache.amazonaws.com/doc/2015-02-02/">
    <DescribeEventsResult>
        <Events>
            <Event>
                <Message>Cache cluster created</Message>
                <SourceType>cache-cluster</SourceType>
                <Date>2015-02-02T18:22:18.202Z</Date>
                <SourceIdentifier>mem01</SourceIdentifier>
            </Event>

 (...output omitted...)

        </Events>
    </DescribeEventsResult>
    <ResponseMetadata>
        <RequestId>e21c81b4-b9cd-11e3-8a16-7978bb24ffdf</RequestId>
    </ResponseMetadata>
</DescribeEventsResponse>
```

Para obtener más información como, por ejemplo, los parámetros disponibles y los valores de parámetros permitidos, consulte `DescribeEvents`.

Notificaciones de eventos y Amazon SNS

ElastiCache puede publicar mensajes con Amazon Simple Notification Service (SNS) cuando se producen eventos significativos en un clúster. Esta característica se puede usar para actualizar las listas de servidor de las máquinas cliente conectadas a puntos de conexión de nodo de caché individuales de un clúster de caché.

> **Note**
>
> Para obtener más información sobre Amazon Simple Notification Service (SNS), incluido la información sobre los precios y enlaces a los documentos de Amazon SNS, consulte la Página de producto de Amazon SNS.

Las notificaciones se publican en un tema específico de Amazon SNS. A continuación se describen los requisitos para las notificaciones:

- Solo se puede configurar un tema para las notificaciones de ElastiCache.

- La cuenta de AWS que tiene la propiedad del tema de Amazon SNS debe ser la misma cuenta que tiene la propiedad del clúster en el que se encuentran habilitadas las notificaciones.
- El tema de Amazon SNS en el que publica no se puede cifrar.

> **Note**
>
> Es posible adjuntar un tema de Amazon SNS cifrado (en reposo) al clúster. Sin embargo, el estado del tema de la consola de ElastiCache se mostrará como inactivo, que desasocia el tema del clúster de forma efectiva cuando ElastiCache envía mensajes al tema.

- El tema de Amazon SNS debe estar en la misma región que el clúster de ElastiCache.

Eventos de ElastiCache

Los siguientes eventos de ElastiCache desencadenan notificaciones de Amazon SNS. Para obtener más información sobre los eventos, consulte Visualización de eventos de ElastiCache.

Nombre del evento	Mensaje	Descripción
ElastiCache:AddCacheNodeComplete	`ElastiCache:AddCacheNodeComplete : cache-cluster`	Se ha agregado un nodo de caché al clúster y ya se puede usar.
ElastiCache:AddCacheNodeFailed debido a direcciones IP libres insuficientes	`ElastiCache:AddCacheNodeFailed : cluster-name`	No se pudo agregar un nodo de caché porque no hay suficientes direcciones IP disponibles.
ElastiCache:CacheClusterParametersChanged	`ElastiCache:CacheClusterParametersChanged : cluster-name`	Se han cambiado uno o varios parámetros del clúster.
ElastiCache:CacheClusterProvisioningComplete	`ElastiCache:CacheClusterProvisioningComplete cluster-name-0001-005`	El aprovisionamiento de un clúster se ha completado y los nodos de caché del clúster de caché están listos para su uso.

Nombre del evento	Mensaje	Descripción
ElastiCache:CacheClusterProvisioningFailed debido a un estado de la red incompatible	`ElastiCache:CacheClusterProvisioningFailed : cluster-name`	Se ha intentado lanzar un nuevo clúster en una nube privada virtual (VPC) que no existe.
ElastiCache:CacheClusterScalingComplete	`CacheClusterScalingComplete : cluster-name`	El escalado del clúster de caché se ha completado con éxito.
ElastiCache:CacheClusterScalingFailed	ElastiCache:CacheClusterScalingFailed : *cluster-name*	Se ha producido un error en la operación de ampliación del clúster de caché.
ElastiCache:CacheClusterSecurityGroupModified	`ElastiCache:CacheClusterSecurityGroupModified : cluster-name`	Se ha producido uno de los eventos siguientes: • La lista de los grupos de seguridad de caché autorizados para el clúster se ha modificado. • Se han autorizado uno o varios de los nuevos grupos de seguridad de EC2 en alguno de los grupos de seguridad de caché asociados al clúster. • Se han rechazado uno o varios de los nuevos grupos de seguridad de EC2 de alguno de los grupos de seguridad de caché asociados al clúster.

Nombre del evento	Mensaje	Descripción
ElastiCache:CacheNodeReplaceStarted	`ElastiCache:CacheNodeReplaceStarted : cluster-name`	ElastiCache detectó que el host que ejecuta un nodo de caché tiene un rendimiento reducido o no se encuentra disponible y comenzó el reemplazo del nodo de caché. **Note** La entrada de DNS del nodo de caché de reemplazo no se ha cambiado. En la mayoría de los casos, cuando se produce este evento, no es necesario actualizar la lista de servidores de sus clientes. Sin embargo, es posible que determinadas bibliotecas de cliente de caché dejen de utilizar el nodo de caché incluso después de que ElastiCache haya reemplazado el nodo de caché. En este caso, la aplicación deberá actualizar la lista de servidores cuando se produzca este evento.

Nombre del evento	Mensaje	Descripción
ElastiCache:CacheNodeReplaceComplete	`ElastiCache:CacheNodeReplaceComplete : cluster-name`	ElastiCache detectó que el host que ejecuta un nodo de caché tiene un desempeño reducido o no se encuentra disponible y completó el reemplazo del nodo de caché. **Note** La entrada de DNS del nodo de caché de reemplazo no se ha cambiado. En la mayoría de los casos, cuando se produce este evento, no es necesario actualizar la lista de servidores de sus clientes. Sin embargo, es posible que determinadas bibliotecas de cliente de caché dejen de utilizar el nodo de caché incluso después de que ElastiCache haya reemplaza do el nodo de caché. En este caso, la aplicación deberá actualizar la lista de servidore s cuando se produzca este evento.

Nombre del evento	Mensaje	Descripción
ElastiCache:CacheNodesRebooted	`ElastiCache:CacheNodesRebooted : cluster-name`	Uno o varios nodos de caché se han reiniciado. Aparece el mensaje (Memcached): `"Cache node %s shutdown"` A continuación, aparece un segundo mensaje: `"Cache node %s restarted"`
ElastiCache:CertificateRenewalComplete (solo Valkey o Redis OSS)	`ElastiCache:CertificateRenewalComplete`	El certificado de entidad de certificación de Amazon se renovó con éxito.
ElastiCache:CreateReplicationGroupComplete	`ElastiCache:CreateReplicationGroupComplete : cluster-name`	El grupo de reproducción se ha creado correctamente.
ElastiCache:DeleteCacheClusterComplete	`ElastiCache:DeleteCacheClusterComplete : cluster-name`	Se ha completado la eliminación de un clúster y de todos los nodos de caché asociados.
ElastiCache:FailoverComplete (solo Valkey o Redis OSS)	`ElastiCache:FailoverComplete : mycluster`	La conmutación por error a un nodo de réplica se ha realizado correctamente.
ElastiCache:ReplicationGroupIncreaseReplicaCountFinished	`ElastiCache:ReplicationGroupIncreaseReplicaCountFinished : cluster-name-0001-005`	Aumentó el número de réplicas en el clúster.

Nombre del evento	Mensaje	Descripción
ElastiCache:ReplicationGroupIncreaseReplicaCountStarted	`ElastiCache:ReplicationGroupIncreaseReplicaCountStarted : cluster-name-0003-004`	Ha comenzado el proceso de agregar réplicas al clúster.
ElastiCache:NodeReplacementCanceled	`ElastiCache:NodeReplacementCanceled : cluster-name`	Un nodo del clúster que cuyo reemplazo estaba programado ya no está programado para el reemplazo.
ElastiCache:NodeReplacementRescheduled	`ElastiCache:NodeReplacementRescheduled : cluster-name`	Un nodo de su clúster que estaba programado para el reemplazo se ha vuelto a programar para el reemplazo durante el nuevo periodo descrito en la notificación. Para obtener información acerca de las acciones que puede emprender, consulte Reemplazo de nodos (Valkey y Redis OSS).
ElastiCache:NodeReplacementScheduled	`ElastiCache:NodeReplacementScheduled : cluster-name`	Un nodo de su clúster se ha programado para el reemplazo durante el periodo que se describe en la notificación. Para obtener información acerca de las acciones que puede emprender, consulte Reemplazo de nodos (Valkey y Redis OSS).

Nombre del evento	Mensaje	Descripción
ElastiCache:RemoveCacheNodeComplete	`ElastiCache:RemoveCacheNodeComplete : cluster-name`	Se ha eliminado un nodo de caché del clúster.
ElastiCache:ReplicationGroupScalingComplete	`ElastiCache:ReplicationGroupScalingComplete : cluster-name`	La operación de ampliación del grupo de reproducción se ha completado correctamente.
ElastiCache:ReplicationGroupScalingFailed	`"Failed applying modification to cache node type to %s."`	Se ha producido un error en la operación de ampliación del grupo de reproducción.
ElastiCache:ServiceUpdateAvailableForNode	`"Service update is available for cache node %s."`	Hay una actualización de autoservicio disponible para el nodo.
ElastiCache:SnapshotComplete (solo Valkey o Redis OSS)	`ElastiCache:SnapshotComplete : cluster-name`	Una instantánea de caché se ha completado correctamente.
ElastiCache:SnapshotFailed (solo Valkey o Redis OSS)	`SnapshotFailed : cluster-name`	Se ha producido un error en la instantánea de caché. Consulte los eventos de caché del clúster para obtener más detalles acerca de la causa. Si describe la instantánea (consulte `DescribeSnapshots`), el estado será `failed`.

Temas relacionados

- Visualización de eventos de ElastiCache

Entrega de registro

> **Note**
>
> La funcionalidad Registro lento es compatible con Valkey 7.x y versiones posteriores, y con los clústeres de Redis OSS y los grupos de replicación que utilicen la versión 6.0 y posteriores del motor.
> El registro del motor es compatible con Valkey 7.x y posteriores y los grupos de replicación y clústeres de Redis OSS que utilicen la versión 6.2 y posteriores del motor.

La entrega de registros le permite transmitir SLOWLOG o Registro del motor a uno de dos destinos:

- Amazon Data Firehose
- Amazon CloudWatch Logs

Puede habilitar y configurar la entrega de registros al crear o modificar un clúster mediante las API de ElastiCache. Cada entrada de registro se entregará al destino especificado en uno de los dos formatos: JSON o TEXT.

Un número fijo de entradas de Registro lento se recuperan de forma periódica del motor. En función del valor especificado para el parámetro `slowlog-max-len` del motor, es posible que las entradas de registro lento adicionales no se entreguen en el destino.

Puede optar por cambiar las configuraciones de entrega o deshabilitar la entrega de registros en cualquier momento mediante la consola de AWS o una de las API de modificación: modify-cache-cluster o modify-replication-group.

Debe establecer el parámetro `apply-immediately` para todas las modificaciones de entrega de registros.

> **Note**
>
> Se aplican cargos de Registros de Amazon CloudWatch cuando la entrega de registros se encuentra habilitada, incluso cuando los registros se entregan directamente a Amazon Data Firehose. Para obtener más información, consulte Registros distribuidos, en Precios de Amazon CloudWatch.

Contenido de una entrada de registro lento

La funcionalidad de Registro lento contiene la siguiente información:

- CacheClusterId: ID del clúster
- CacheNodeId: el ID del nodo de caché
- Id: una identificación progresiva única para cada entrada de registro lento
- Marca temporal: la marca de tiempo de Unix en la que se procesó el comando registrado
- Duración: la cantidad de tiempo necesario para la ejecución, en microsegundos
- Comando: el comando utilizado por el cliente Por ejemplo, `set foo bar` donde `foo` es la clave y `bar` es el valor. ElastiCache reemplaza el nombre y el valor de la clave real por `(2 more arguments)` para evitar exponer información confidencial.
- ClientAddress: la dirección IP y el puerto del cliente
- ClientName: el nombre del cliente si se establece a través del comando `CLIENT SETNAME`

Contenido de una entrada de registro del motor

El registro del motor de ElastiCache incluye la siguiente información:

- CacheClusterId: ID del clúster
- CacheNodeId: el ID del nodo de caché
- Nivel de registro: LogLevel puede ser uno de los siguientes:`VERBOSE("-")`, `NOTICE("*")`, `WARNING("#")`.
- Tiempo: la hora UTC del mensaje registrado. La hora tiene el siguiente formato: `"DD MMM YYYY hh:mm:ss.ms UTC"`
- Rol: rol del nodo desde el que se emite el registro. Puede ser uno de los siguientes: «M» para principal, «S» para réplica, «C» para proceso secundario de escritor que trabaja en RDB/AOF o «X» para centinela.
- Mensaje: mensaje de registro del motor.

Permisos para configurar el registro

Debe incluir los siguientes permisos de IAM en la política de usuario/rol de IAM:

- `logs:CreateLogDelivery`

- `logs:UpdateLogDelivery`
- `logs:DeleteLogDelivery`
- `logs:GetLogDelivery`
- `logs:ListLogDeliveries`

Para obtener más información, consulte Información general sobre los permisos y las políticas de la administración del acceso.

Especificaciones del tipo y formato de registro

Registro lentos

El registro lento admite JSON y TEXT

A continuación se muestra un ejemplo de formato JSON:

```
{
  "CacheClusterId": "logslowxxxxmsxj",
  "CacheNodeId": "0001",
  "Id": 296,
  "Timestamp": 1605631822,
  "Duration (us)": 0,
  "Command": "GET ... (1 more arguments)",
  "ClientAddress": "192.168.12.104:55452",
  "ClientName": "logslowxxxxmsxj##"
}
```

A continuación se muestra un ejemplo de formato TEXT:

```
logslowxxxxmsxj,0001,1605631822,30,GET ... (1 more
 arguments),192.168.12.104:55452,logslowxxxxmsxj##
```

Registro del motor

El registro del motor admite JSON y TEXT

A continuación se muestra un ejemplo de formato JSON:

```
{
```

```
  "CacheClusterId": "xxxxxxxxxzy-engine-log-test",
  "CacheNodeId": "0001",
  "LogLevel": "VERBOSE",
  "Role": "M",
  "Time": "12 Nov 2020 01:28:57.994 UTC",
  "Message": "Replica is waiting for next BGSAVE before synchronizing with the primary.
 Check back later"
}
```

A continuación se muestra un ejemplo de formato TEXT:

```
xxxxxxxxxxxzy-engine-log-test/0001:M 29 Oct 2020 20:12:20.499 UTC * A slow-running Lua
 script detected that is still in execution after 1000 microseconds.
```

Destinos de registro de ElastiCache

En esta sección se describen los destinos de registro que puede elegir para los registros de ElastiCache. Cada sección proporciona instrucciones a fin de configurar el registro para el tipo de destino e información sobre cualquier comportamiento específico del tipo de destino. Después de configurar el destino de registro, puede agregar especificaciones a la configuración de registro de ElastiCache para iniciar el registro en él.

Temas

- Amazon CloudWatch Logs
- Amazon Data Firehose

Amazon CloudWatch Logs

- Puede especificar un grupo de CloudWatch Logs donde se entregarán los registros.
- Los registros de varios clústeres de Valkey o Redis OSS y grupos de replicación se pueden entregar al mismo grupo de registros.
- Se creará un flujo de registro nuevo para cada nodo dentro de un clúster o grupo de reproducción y los registros se entregarán a los flujos de registro respectivos. El nombre de la secuencia de registro utilizará el siguiente formato: `elasticache/${engine-name}/${cache-cluster-id}/${cache-node-id}/${log-type}`

Permisos para publicar registros en CloudWatch Logs

Debe tener la siguiente configuración de permisos para configurar ElastiCache para enviar registros a un grupo de registros de Registros de CloudWatch:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:CreateLogDelivery",
                "logs:GetLogDelivery",
                "logs:UpdateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:ListLogDeliveries"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "ElastiCacheLogging"
        },
        {
            "Sid": "ElastiCacheLoggingCWL",
            "Action": [
                "logs:PutResourcePolicy",
                "logs:DescribeResourcePolicies",
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow"
        }
    ]
}
```

Para obtener más información, consulte Registros enviados a CloudWatch Logs.

Amazon Data Firehose

- Puede especificar un flujo de entrega de Firehose en el que se entregarán los registros.
- Los registros de varios grupos de replicación y clústeres de Valkey o Redis OSS se pueden entregar al mismo flujo de entrega.
- Los registros de cada nodo dentro de un clúster o grupo de reproducción se entregarán al mismo flujo de entrega. Puede distinguir los mensajes de registro de diferentes nodos de caché en función de los `cache-cluster-id` y `cache-node-id` incluidos en cada mensaje de registro.
- En la región Asia-Pacífico (Osaka), la entrega de registros de Firehose no se encuentra disponible.

Permisos para publicar registros en Firehose

Para configurar ElastiCache para que envíe registros a un flujo de entrega de Amazon Kinesis Data Firehose debe contar con los siguientes permisos.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:CreateLogDelivery",
                "logs:GetLogDelivery",
                "logs:UpdateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:ListLogDeliveries"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "ElastiCacheLogging"
        },
        {
            "Sid": "ElastiCacheLoggingFHSLR",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "*",
```

```
            "Effect": "Allow"
        },
        {
            "Sid": "ElastiCacheLoggingFH",
            "Action": [
                "firehose:TagDeliveryStream"
            ],
            "Resource": "arn:aws:iam::*:role/*",
            "Effect": "Allow"
        }
    ]
}
```

Especificación de la entrega de registros mediante la consola

Mediante el uso de Consola de administración de AWS, puede crear un clúster de Valkey o Redis OSS (modo de clúster deshabilitado) siguiendo los pasos de Creación de un clúster de Valkey (modo de clúster deshabilitado) (consola) o puede crear un clúster de Valkey o Redis OSS (modo de clúster habilitado) siguiendo los pasos de Creación de un clúster de Valkey o Redis OSS (modo de clúster habilitado) (consola). En cualquier caso, puede configurar la entrega de registros al realizar lo siguiente:

1. En Configuración avanzada, elija Registros y, a continuación, marque Registros lentos o Registros del motor.
2. En Log format (Formato de registro), elija Text o JSON.
3. En Destination Type (Tipo de destino), elija CloudWatch Logs o Kinesis Firehose.
4. En Log destination (Destino de registro), elija Create new (Crear nuevo) e ingrese el nombre del bucket de Amazon S3, el nombre del grupo de registro de CloudWatchLogs o el nombre del flujo de Kinesis Data Firehose, o elija Select existing (Seleccionar existente) y, a continuación, elija el nombre del grupo de registros de CloudWatchLogs o el nombre del flujo de Kinesis Data Firehose.

Cuándo modificar un clúster:

Puede elegir entre habilitar o deshabilitar la entrega de registros o cambiar el tipo de destino, el formato o el destino:

1. Inicie sesión en la consola y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. En el panel de navegación, elija Clústeres de Valkey o Clústeres de Redis OSS.
3. En la lista de clústeres, elija el clúster que desea modificar. Elija el Cluster name (Nombre del clúster) y no la casilla de verificación al lado.
4. En la página de Cluster name (Nombre del clúster), seleccione la pestaña de Logs (Registros).
5. Para habilitar/deshabilitar los registros lentos, elija Enable slow logs (Habilitar registros lentos) o Disable slow logs (Deshabilitar registros lentos).
6. Para habilitar/deshabilitar los registros del motor, elija Enable engine logs (Habilitar registros del motor) o Disable engine logs (Deshabilitar registros del motor).
7. Para cambiar la configuración, elija Modify slow logs (Modificar registros lentos) o Modify engine logs (Mdificar registros del motor):
   - En Destination Type (Tipo de destino), elija CloudWatch Logs o Kinesis Firehose.
   - En Log destination (Destino del registro), elija Create new (Crear nuevo) e ingrese el nombre del grupo de registros de CloudWatch Logs o el nombre de la secuencia de Kinesis Data Firehose. O elija Select existing (Seleccionar existente) y, a continuación, elija el nombre del grupo de registros de CloudWatch Logs o el nombre de la secuencia de Kinesis Data Firehose.

Especificación de la entrega de registros mediante la AWS CLI

Registro lento

Cree un grupo de reproducción con entrega de registro lento a CloudWatch Logs.

Para Linux, macOS o Unix:

```
aws elasticache create-replication-group \
    --replication-group-id test-slow-log \
    --replication-group-description test-slow-log \
    --engine redis \
    --cache-node-type cache.r5.large \
    --num-cache-clusters 2 \
    --log-delivery-configurations '{
        "LogType":"slow-log",
        "DestinationType":"cloudwatch-logs",
```

```
        "DestinationDetails":{
          "CloudWatchLogsDetails":{
            "LogGroup":"my-log-group"
          }
        },
        "LogFormat":"json"
      }'
```

Para Windows:

```
aws elasticache create-replication-group ^
    --replication-group-id test-slow-log ^
    --replication-group-description test-slow-log ^
    --engine redis ^
    --cache-node-type cache.r5.large ^
    --num-cache-clusters 2 ^
    --log-delivery-configurations '{
        "LogType":"slow-log",
        "DestinationType":"cloudwatch-logs",
        "DestinationDetails":{
          "CloudWatchLogsDetails":{
            "LogGroup":"my-log-group"
          }
        },
        "LogFormat":"json"
      }'
```

Modificar un grupo de reproducción para entregar un registro lento a CloudWatch Logs

Para Linux, macOS o Unix:

```
aws elasticache modify-replication-group \
    --replication-group-id test-slow-log \
    --apply-immediately \
    --log-delivery-configurations '
    {
      "LogType":"slow-log",
      "DestinationType":"cloudwatch-logs",
      "DestinationDetails":{
        "CloudWatchLogsDetails":{

          "LogGroup":"my-log-group"
        }
```

```
      },
      "LogFormat":"json"
    }'
```

Para Windows:

```
aws elasticache modify-replication-group ^
    --replication-group-id test-slow-log ^
    --apply-immediately ^
    --log-delivery-configurations '
    {
      "LogType":"slow-log",
      "DestinationType":"cloudwatch-logs",
      "DestinationDetails":{
        "CloudWatchLogsDetails":{
          "LogGroup":"my-log-group"
        }
      },
      "LogFormat":"json"
    }'
```

Modificar un grupo de reproducción para deshabilitar la entrega de registro lento

Para Linux, macOS o Unix:

```
aws elasticache modify-replication-group \
    --replication-group-id test-slow-log \
    --apply-immediately \
    --log-delivery-configurations '
    {
      "LogType":"slow-log",
      "Enabled":false
    }'
```

Para Windows:

```
aws elasticache modify-replication-group ^
    --replication-group-id test-slow-log ^
    --apply-immediately ^
    --log-delivery-configurations '
    {
      "LogType":"slow-log",
```

```
      "Enabled":false
    }'
```

Registro del motor

Cree un grupo de reproducción con entrega de registro del motor a CloudWatch Logs.

Para Linux, macOS o Unix:

```
aws elasticache create-replication-group \
    --replication-group-id test-slow-log \
    --replication-group-description test-slow-log \
    --engine redis \
    --cache-node-type cache.r5.large \
    --num-cache-clusters 2 \
    --log-delivery-configurations '{
        "LogType":"engine-log",
        "DestinationType":"cloudwatch-logs",
        "DestinationDetails":{
          "CloudWatchLogsDetails":{
            "LogGroup":"my-log-group"
          }
        },
        "LogFormat":"json"
      }'
```

Para Windows:

```
aws elasticache create-replication-group ^
    --replication-group-id test-slow-log ^
    --replication-group-description test-slow-log ^
    --engine redis ^
    --cache-node-type cache.r5.large ^
    --num-cache-clusters 2 ^
    --log-delivery-configurations '{
        "LogType":"engine-log",
        "DestinationType":"cloudwatch-logs",
        "DestinationDetails":{
          "CloudWatchLogsDetails":{
            "LogGroup":"my-log-group"
          }
        },
        "LogFormat":"json"
```

```
      }'
```

Modifique un grupo de replicación para entregar un registro del motor a Firehose

Para Linux, macOS o Unix:

```
aws elasticache modify-replication-group \
    --replication-group-id test-slow-log \
    --apply-immediately \
    --log-delivery-configurations '
    {
      "LogType":"engine-log",
      "DestinationType":"kinesis-firehose",
      "DestinationDetails":{
      "KinesisFirehoseDetails":{
         "DeliveryStream":"test"
       }
     },
      "LogFormat":"json"
    }'
```

Para Windows:

```
aws elasticache modify-replication-group ^
    --replication-group-id test-slow-log ^
    --apply-immediately ^
    --log-delivery-configurations '
    {
      "LogType":"engine-log",
      "DestinationType":"kinesis-firehose",
      "DestinationDetails":{
      "KinesisFirehoseDetails":{
         "DeliveryStream":"test"
       }
      },
      "LogFormat":"json"
    }'
```

Modifique un grupo de reproducción para cambiar al formato del motor

Para Linux, macOS o Unix:

```
aws elasticache modify-replication-group \
```

```
    --replication-group-id test-slow-log \
    --apply-immediately \
    --log-delivery-configurations '
    {
       "LogType":"engine-log",
       "LogFormat":"json"
    }'
```

Para Windows:

```
aws elasticache modify-replication-group ^
    --replication-group-id test-slow-log ^
    --apply-immediately ^
    --log-delivery-configurations '
    {
       "LogType":"engine-log",
       "LogFormat":"json"
    }'
```

Modifique un grupo de reproducción para deshabilitar la entrega de registro de motor

Para Linux, macOS o Unix:

```
aws elasticache modify-replication-group \
    --replication-group-id test-slow-log \
    --apply-immediately \
    --log-delivery-configurations '
    {
      "LogType":"engine-log",
      "Enabled":false
    }'
```

Para Windows:

```
aws elasticache modify-replication-group ^
    --replication-group-id test-slow-log ^
    --apply-immediately ^
    --log-delivery-configurations '
    {
      "LogType":"engine-log",
      "Enabled":false
    }'
```

Monitoreo del uso con métricas de CloudWatch

ElastiCache proporciona métricas que permiten monitorear los clústeres. Puede acceder a dichas métricas a través de CloudWatch. Para obtener más información sobre CloudWatch, consulte la documentación de CloudWatch.

ElastiCache proporciona métricas de nivel de anfitrión (por ejemplo, sobre el uso de la CPU) y métricas específicas del software del motor de caché (por ejemplo, las tasas de caché y los errores de caché). Estas métricas se miden y publican para cada nodo de caché en intervalos de 60 segundos.

> **Important**
>
> Considere la posibilidad de configurar alarmas de CloudWatch con determinadas métricas clave para, de este modo, recibir una notificación si el rendimiento del clúster comienza a disminuir. Para obtener más información, consulte la sección ¿Qué métricas debo monitorear? de esta guía.

Temas

- Métricas de nivel de host
- Métricas de Valkey y Redis OSS
- Métricas de Memcached
- ¿Qué métricas debo monitorear?
- Elección de periodos y estadísticas de métricas
- Monitoreo de métricas de nodo y clúster de CloudWatch

Métricas de nivel de host

El espacio de nombres `AWS/ElastiCache` incluye las siguientes métricas de nivel de host para los distintos nodos de caché. Estas métricas se miden y publican para cada nodo de caché en intervalos de 60 segundos.

Véase también

- Métricas de Valkey y Redis OSS

Métrica	Descripción	Unidad
`CPUUtilization`	El porcentaje de uso de la CPU para todo el host. Dado que Valkey y Redis OSS utilizan un solo subproceso, recomendamos que supervise la métrica `EngineCPUUtilizati` on de los nodos con cuatro o más vCPU.	Porcentaje
`CPUCreditBalance`	La cantidad de créditos de la CPU obtenidos que una instancia ha acumulado desde que se lanzó o se inició. Para T2 Standard, CPUCredit Balance incluye además el número de créditos de lanzamiento que se acumularon. Los créditos se acumulan en el saldo de créditos una vez obtenidos y se eliminan del saldo de créditos cuando se gastan. El saldo de créditos tiene un límite máximo, determina do por el tamaño de la instancia. Una vez que se ha alcanzado el límite, los nuevos créditos obtenidos se descartarán. Para T2 Standard, los créditos de inicialización no cuentan para el límite. Los créditos de CPUCreditBalance se encuentran disponibles para que la instancia los gaste y aumente la utilización de la CPU por encima de la referencia. Las métricas de créditos de CPU solo están disponibles cada cinco minutos. Estas métricas no se encuentran disponibles para las instancias de rendimiento ampliable T2.	Créditos (vCPU/min utos)

Métrica	Descripción	Unidad
`CPUCreditUsage`	La cantidad de créditos de CPU gastados por la instancia para la utilización de la CPU. Un crédito de CPU equivale a una vCPU ejecutándose al 100% de utilización durante un minuto o una combinación equivalente de unidades de vCPU, utilización y tiempo (por ejemplo, una vCPU ejecutándose al 50% durante dos minutos o dos vCPU ejecutándose al 25% durante dos minutos). Las métricas de créditos de CPU solo están disponibles cada cinco minutos. Si se especifica un periodo superior a cinco minutos, utilice la estadística Sum en lugar de Average. Estas métricas no se encuentran disponibles para las instancias de rendimiento ampliable T2.	Créditos (vCPU/minutos)
`FreeableMemory`	La cantidad de memoria libre disponible en el host. Esto se deriva de la RAM, los búferes y la caché que el sistema operativo reporta como liberables.	Bytes
`NetworkBytesIn`	El número de bytes que el host ha leído de la red.	Bytes
`NetworkBytesOut`	El número de bytes enviados en todas las interfaces de red por la instancia.	Bytes
`NetworkPacketsIn`	El número de paquetes recibidos en todas las interfaces de red por la instancia. Esta métrica identifica el volumen de tráfico de red entrante en cuanto al número de paquetes de una sola instancia.	Recuento

Métrica	Descripción	Unidad
NetworkPacketsOut	El número de paquetes enviados en todas las interfaces de red por la instancia. Esta métrica identifica el volumen de tráfico de red saliente en cuanto al número de paquetes de una sola instancia.	Recuento
NetworkBandwidthInAllowanceExceeded	El número de paquetes formados en cola o eliminados el ancho de banda agregado entrante superó el máximo de la instancia.	Recuento
NetworkConntrackAllowanceExceeded	El número de paquetes eliminados porque el seguimiento de conexiones superó el máximo de la instancia y no se pudieron establecer nuevas conexiones. Esto puede provocar la pérdida de paquetes para el tráfico hacia o desde la instancia.	Recuento
NetworkBandwidthOutAllowanceExceeded	El número de paquetes en cola o eliminados porque el ancho de banda agregado saliente superó el máximo de la instancia.	Recuento
NetworkPacketsPerSecondAllowanceExceeded	El número de paquetes en cola o eliminados debido a que los paquetes bidireccional superaron el máximo de la instancia.	Recuento
NetworkMaxBytesIn	La ráfaga por segundo máxima de bytes recibidos cada minuto.	Bytes
NetworkMaxBytesOut	La ráfaga por segundo máxima de bytes transmitidos cada minuto.	Bytes
NetworkMaxPacketsIn	La ráfaga por segundo máxima de paquetes recibidos cada minuto.	Recuento
NetworkMaxPacketsOut	La ráfaga por segundo máxima de paquetes transmitidos cada minuto.	Recuento

Métrica	Descripción	Unidad
SwapUsage	La cantidad de espacio de intercambio utilizado en el host.	Bytes

Métricas de Valkey y Redis OSS

El espacio de nombres Amazon ElastiCache incluye las siguientes métricas de Valkey y Redis OSS. Estas métricas son las mismas cuando se utiliza el motor de Valkey.

Con la excepción de ReplicationLag, EngineCPUUtilization, SuccessfulWriteRequestLatency y SuccessfulReadRequestLatency, estas métricas se obtienen del comando info. Cada métrica se calcula en el nivel de nodo de caché.

Para ver documentación completa del comando info, consulte http://valkey.io/commands/info.

Véase también

- Métricas de nivel de host

Métrica	Descripción	Unidad
ActiveDefragHits	El número de reasignaciones de valor por minuto que ha realizado el proceso de desfragmentación activo. Se obtiene de la estadística de active_defrag_hits en INFO.	Número
AuthenticationFailures	Número total de intentos fallidos de autenticarse en Valkey o Redis OSS mediante el comando AUTH. Puede encontrar más información sobre los errores de autenticación individuales mediante el comando ACL LOG. Sugerimos configurar una alarma para detectar intentos de acceso sin autorización.	Recuento

Métrica	Descripción	Unidad
BytesUsedForCache	Número total de bytes asignados por Valkey o Redis OSS para todos los propósitos, incluido los conjuntos de datos, los búferes, etc.	Bytes
	Dimension: Tier=Memory para clústeres de Valkey o Redis OSS mediante Organización de datos en niveles en ElastiCache: número total de bytes utilizados para la caché por memoria. Este es el valor de la estadística de used_memory en INFO.	Bytes
	Dimension: Tier=SSD para clústeres de Valkey o Redis OSS mediante Organización de datos en niveles en ElastiCache: número total de bytes utilizados para la caché por SSD.	Bytes
BytesReadFromDisk	Número total de bytes leídos del disco por minuto. Compatible solo con clústeres que utilizan Organización de datos en niveles en ElastiCache.	Bytes
BytesWrittenToDisk	Número total de bytes escritos en el disco por minuto. Compatible solo con clústeres que utilizan Organización de datos en niveles en ElastiCache.	Bytes
CacheHits	El número de búsquedas de claves solo de lectura realizadas correctamente en el diccionario principal. Se obtiene de la estadística de keyspace_hits en INFO.	Recuento
CacheMisses	El número de búsquedas de claves solo de lectura que no se realizaron correctamente en el diccionario principal. Se obtiene de la estadística de keyspace_misses en INFO.	Recuento

Métrica	Descripción	Unidad
`CommandAuthorizationFailures`	Número total de intentos fallidos de los usuarios de ejecutar comandos a los que no tienen permiso para llamar. Puede encontrar más información sobre los errores de autenticación individuales mediante el comando ACL LOG. Sugerimos configurar una alarma para detectar intentos de acceso sin autorización.	Recuento
`CacheHitRate`	Indica la eficiencia de uso de la instancia de Valkey o Redis OSS. Si la relación de caché es inferior a aproximadamente 0,8, significa que una cantidad significativa de claves se han expulsado, han caducado o no existen. Esto se calcula utilizando estadísticas `cache_hits` y `cache_misses` de la siguiente manera: `cache_hits /(cache_hits + cache_misses)` .	Porcentaje
`ChannelAuthorizationFailures`	El número total de intentos con error de los usuarios de acceder a canales a los que no tienen permiso para acceder. Puede encontrar más información sobre los errores de autenticación individuales mediante el comando ACL LOG. Sugerimos configurar una alarma en esta métrica para detectar intentos de acceso sin autorización.	Recuento
`CurrConnections`	Número de conexiones de cliente, excluido las conexiones de réplicas de lectura. ElastiCache utiliza de 4 a 6 de las conexiones para supervisar el clúster en cada caso. Se obtiene de la estadística de `connected_clients` en INFO.	Recuento

Métrica	Descripción	Unidad
`CurrItems`	El número de elementos en la caché. Se obtiene de la estadística de `keyspace`, sumando todas las claves del espacio de claves completo.	Recuento
	`Dimension: Tier=Memory` para clústeres que utilizan Organización de datos en niveles en ElastiCache. Número de elementos en la memoria.	Recuento
	`Dimension: Tier=SSD` (unidades de estado sólido) para clústeres que utilizan Organización de datos en niveles en ElastiCache. Número de elementos en SSD.	Recuento
`CurrVolatileItems`	Número total de claves en todas las bases de datos que tienen un conjunto ttl. Se obtiene de la estadística de `expires`, sumando todas las claves con un conjunto ttl en el espacio de claves completo.	Recuento
`DatabaseCapacityUsagePercentage`	El porcentaje de la capacidad de datos total para el clúster que está en uso. En las instancias con niveles de datos, la métrica se calcula como `(used_memory - mem_not_counted_for_evict + SSD used) / (maxmemory + SSD total capacity)` , donde `used_memory` y `maxmemory` se toman de INFO. En el resto de los casos, la métrica se calcula utilizando `used_memory/maxmemory` .	Porcentaje

Métrica	Descripción	Unidad

`DatabaseCapacityUsageCountedForEvictPercentage`	El porcentaje de la capacidad de datos total para el clúster que está en uso, excluida la memoria utilizada para gastos generales y COB. Esta métrica se calcula de la siguiente manera:
	
	`used_memory - mem_not_counted_for_evict/maxmemory`
	
	En las instancias de datos por niveles, la métrica se calcula de la siguiente manera:
	
	`(used_memory + SSD used) / (maxmemory + SSD total capacity)`
	
	donde `used_memory` y `maxmemory` se toman de INFO
		Porcentaje

`DatabaseMemoryUsagePercentage`	El porcentaje de la memoria para el clúster que está en uso. Esto se calcula usando `used_memory/maxmemory` de INFO.	Porcentaje
`DatabaseMemoryUsageCountedForEvictPercentage`	El porcentaje de la memoria para el clúster que está en uso, excluida la memoria utilizada para gastos generales y COB. Esto se calcula usando `used_memory-mem_not_counted_for_evict/maxmemory` de INFO.	Porcentaje

Métrica	Descripción	Unidad
DB0AverageTTL	Expone avg_ttl de DBO a partir de la estadística de keyspace del comando INFO. Las réplicas no hacen que venzan las claves, sino que esperan a que los nodos principal es hagan que venzan las claves. Cuando un nodo principal hace que una clave venza (o la expulsa debido a LRU), sintetiza un comando DEL, que se transmite a todas las réplicas. Por lo tanto, DB0AverageTTL es 0 para los nodos de réplica, debido al hecho de que no vencen las claves y, por lo tanto, no realizan un seguimiento de TTL.	Milisegundos

Métrica	Descripción	Unidad
`EngineCPUUtilization`	Proporciona el uso de la CPU del subproceso del motor de Valkey o Redis OSS. Como Valkey y Redis OSS utilizan un solo subproceso, puede utilizar esta métrica para analizar la carga del propio proceso. La métrica `EngineCPUUtilization` brinda una visibilidad más precisa del proceso. Puede utilizarla junto con la métrica `CPUUtilization` . `CPUUtilization` muestra el uso de la CPU para la instancia de servidor como un conjunto, lo que incluye otro sistema operativo y los procesos de administración. En los tipos de nodos grandes que tienen cuatro o más vCPU, utilice la métrica `EngineCPUUtilization` para monitorear y establecer umbrales de escalado. **Note** En un host de ElastiCache existen procesos en segundo plano que monitorean el host para proporcionar una experiencia de base de datos administrada. Estos procesos en segundo plano pueden ocupar una parte importante de la carga de trabajo de la CPU. Esto no es significativo en host más grandes con más de 2 vCPU. Pero puede afectar a hosts más pequeños con 2 vCPU o menos. Si solo supervisa la métrica `EngineCPUUtilization` , no tendrá constancia de las situaciones en las que el host esté sobrecargado con un alto uso de CPU de Valkey o Redis OSS y un	Porcentaje

Métrica	Descripción	Unidad
	alto uso de CPU de los procesos de supervisión en segundo plano. Por lo tanto, recomendamos monitorear la métrica `CPUUtilization` en los hosts con 2 vCPU o menos.	
`Evictions`	El número de claves que se han desalojado debido al límite `maxmemory`. Se obtiene de la estadística de `evicted_keys` en INFO.	Recuento
`GlobalDatastoreReplicationLag`	Es el retraso entre el nodo primario de la región secundaria y el nodo primario de la región principal. Para Valkey o Redis OSS en modo de clúster habilitado, el retraso indica el retardo máximo entre las particiones.	Segundos
`IamAuthenticationExpirations`	El número total de conexiones de Valkey o Redis OSS autenticadas por IAM que han caducado. Puede encontrar más información sobre Autenticación con IAM en la guía del usuario.	Recuento
`IamAuthenticationThrottling`	El número total de solicitudes AUTH o HELLO de Valkey o Redis OSS autenticadas por IAM con limitación. Puede encontrar más información sobre Autenticación con IAM en la guía del usuario.	Recuento
`IsMaster`	Indica si el nodo es el nodo primario de la partición o clúster actual. La métrica puede ser 0 (no principal) o 1 (principal).	Recuento

Métrica	Descripción	Unidad
KeyAuthorizationFailures	Número total de intentos fallidos de los usuarios de acceder a claves a las que no tienen permiso para acceder. Puede encontrar más información sobre los errores de autenticación individuales mediante el comando ACL LOG. Sugerimos configurar una alarma para detectar intentos de acceso sin autorización.	Recuento
KeysTracked	Número de claves a las que Valkey o Redis OSS realiza un seguimiento como un porcentaje de tracking-table-max-keys . El seguimiento de claves se utiliza para ayudar al almacenamiento en caché del lado del cliente y notifica a los clientes cuando se modifican las claves.	Recuento
MemoryFragmentationRatio	Indica la eficiencia en la asignación de memoria del motor de Valkey o Redis OSS. Determinados umbrales supondrán comportamientos diferentes. El valor recomendado es tener fragmentación por encima de 1,0. Esto se calcula a partir de mem_fragmentation_ratio statistic de INFO.	Número

Métrica	Descripción	Unidad
`NewConnections`	El número total de conexiones que ha aceptado el servidor durante este periodo. Se obtiene de la estadística de `total_connections_received` en INFO. **Note** Si utiliza ElastiCache para Redis OSS versión 5 o inferior, ElastiCache utiliza entre dos y cuatro de las conexione s notificadas por esta métrica para supervisar el clúster. Sin embargo, cuando se utiliza ElastiCache para Redis OSS versión 6 o superior, las conexiones utilizadas por ElastiCac he para supervisar el clúster no se incluyen en esta métrica.	Recuento
`NumItemsReadFromDisk`	El número total de elementos recuperados del disco por minuto. Compatible solo con clústeres que utilizan Organización de datos en niveles en ElastiCache.	Recuento
`NumItemsWrittenToDisk`	El número total de elementos escritos en disco por minuto. Compatible solo con clústeres que utilizan Organización de datos en niveles en ElastiCache.	Recuento
`MasterLinkHealthStatus`	Este estado tiene dos valores: 0 o 1. El valor 0 indica que los datos del nodo principal de ElastiCache no se encuentran sincronizados con Valkey o Redis OSS en EC2. El valor 1 indica que los datos están sincronizados. Para completar la migración, utilice la operación de la API CompleteMigration.	Booleano

Métrica	Descripción	Unidad
`Reclaimed`	El número total de eventos de vencimiento de clave. Se obtiene de la estadística de `expired_keys` en INFO.	Recuento
`ReplicationBytes`	Para los nodos en una configuración que se replica, `ReplicationBytes` indica el número de bytes que el nodo primario envía a todas las réplicas. Esta métrica representa la carga de escritura del grupo de reproducción. Se obtiene de la estadística de `master_repl_offset` en INFO.	Bytes
`ReplicationLag`	Esta métrica solo se aplica a un nodo que se ejecuta como una réplica de lectura. Representa lo que tarda la réplica en aplicar los cambios del nodo principal, en segundos. Desde Valkey 7.2 en adelante y el motor de Redis OSS versión 5.0.6 en adelante, el retraso puede medirse en milisegundos.	Segundos
`SaveInProgress`	Esta métrica binaria devuelve 1 cuando hay una operación de guardado en segundo plano en curso (con ramificación o sin ella) y 0 en caso contrario. El proceso de guardado en segundo plano se usa normalmente durante las instantáneas y las sincronizaciones. Estas operaciones pueden mermar el rendimiento. Con la métrica `SaveInProgress`, puede diagnosticar si un proceso de guardado en segundo plano causó o no una merma del rendimiento. Se obtiene de la estadística de `rdb_bgsave_in_progress` en INFO.	Booleano

Métrica	Descripción	Unidad
`TrafficManagementActive`	Indica si ElastiCache para Redis OSS administra el tráfico de forma activa ajustando el tráfico asignado a los comandos entrantes, la supervisión o la replicación. El tráfico se administra cuando se envían más comandos al nodo de los que Valkey o Redis OSS pueden procesar y se utiliza para mantener la estabilidad y el funcionamiento óptimo del motor. Cualquier punto de datos de 1 puede indicar que ha disminuido la escala del nodo para la carga de trabajo que se proporciona. **Note** Si esta métrica permanece activa, evalúe el clúster para decidir si es necesario escalar verticalmente o escalar horizontalmente. Las métricas relacionadas incluyen `NetworkBandwidthOutAllowanceExceeded` y `EngineCPUUtilization`.	Booleano
`SuccessfulWriteRequestLatency`	Latencia de las solicitudes de escritura correctas. Estadísticas válidas: Average, Sum, Min, Max, Sample Count y cualquier percentil entre p0 y p100. El recuento de muestras solo incluye los comandos que se han ejecutado correctamente.	Microsegundos

Métrica	Descripción	Unidad
SuccessfulReadRequestLatency	Latencia de las solicitudes de lectura correctas. Estadísticas válidas: Average, Sum, Min, Max, Sample Count y cualquier percentil entre p0 y p100. El recuento de muestras solo incluye los comandos que se han ejecutado correctamente.	Microsegundos
ErrorCount	Número total de comandos fallidos durante el periodo de tiempo especificado. Estadísticas válidas: Average, Sum, Min y Max	Recuento
SearchNumberOfIndexes	Número de índices creados	Recuento
SearchTotalIndexedDocuments	Número total de claves en todos los índices	Recuento
SearchUsedMemoryBytes	Número de bytes de memoria consumidos en todas las estructuras de datos de búsqueda	Bytes

Disponibilidad de EngineCPUUtilization

AWSLas regiones de que se indican a continuación se encuentran disponibles en todos los tipos de nodos admitidos.

Región	Nombre de la región
us-east-2	Este de EE. UU. (Ohio)
us-east-1	Este de EE. UU. (Norte de Virginia)
us-west-1	Oeste de EE. UU. (Norte de California)
us-west-2	Oeste de EE. UU. (Oregón)

Región	Nombre de la región
ap-northeast-1	Asia-Pacífico (Tokio)
ap-northeast-2	Asia-Pacífico (Seúl)
ap-northeast-3	Asia-Pacífico (Osaka)
ap-east-1	Asia-Pacífico (Hong Kong)
ap-south-1	Asia-Pacífico (Mumbai)
ap-southeast-1	Asia-Pacífico (Singapur)
ap-southeast-2	Asia-Pacífico (Sídney)
ap-southeast-3	Asia-Pacífico (Yakarta)
ca-central-1	Canadá (centro)
cn-north-1	China (Pekín)
cn-northwest-2	China (Ningxia)
me-south-1	Medio Oriente (Baréin)
eu-central-1	Europa (Fráncfort)
eu-west-1	Europa (Irlanda)
eu-west-2	Europa (Londres)
eu-west-3	UE (París)
eu-south-1	Europa (Milán)
af-south-1	África (Ciudad del Cabo)
eu-north-1	Europa (Estocolmo)
sa-east-1	América del Sur (São Paulo)

Región	Nombre de la región
us-gov-west-1	AWS GovCloud (EE. UU. Oeste)
us-gov-east-1	AWS GovCloud (Este de EE. UU.)

A continuación se muestran agrupaciones de determinados tipos de comandos, que se obtienen de : info commandstats. La sección commandstats proporciona estadísticas basadas en el tipo de comando, incluido el número de llamadas, el tiempo total de CPU consumido por estos comandos y el promedio de CPU consumido por ejecución de comando. Para cada tipo de comando, se agrega la siguiente línea: `cmdstat_XXX: calls=XXX,usec=XXX,usec_per_call=XXX`.

Las métricas de latencia que se indican a continuación se calculan con la estadística de commandstats de INFO. Se calculan de la siguiente manera: `delta(usec)/delta(calls)`. `delta` se calcula como la diferencia en un minuto. La latencia se define como el tiempo de CPU que ElastiCache tarda en procesar el comando. Tenga en cuenta que para los clústeres que utilizan la organización de datos en niveles, el tiempo que tarda en obtener elementos de SSD no se incluye en estas mediciones.

Para obtener una lista completa de los comandos disponibles, consulte los comandos en la documentación de Valkey.

Métrica	Descripción	Unidad
`ClusterBasedCmds`	El número total de comandos basados en clústeres. Se obtiene de la estadística de `commandstats` sumando todos los comandos que actúan en un clúster (`cluster slot`, `cluster info`, etc.).	Recuento
`ClusterBasedCmdsLatency`	Latencia de comandos basados en clústeres.	Microsegundos
`EvalBasedCmds`	El número total de comandos para los comandos basados en eval. Se obtiene de la estadística de `commandstats` sumando eval, evalsha.	Recuento

Métrica	Descripción	Unidad
EvalBasedCmdsLatency	Latencia de comandos basados en eval.	Microsegundos
GeoSpatialBasedCmds	Número total de comandos para comandos basados en condiciones geoespaciales. Esto se obtiene de la estadística de commandstats . Esto se obtiene al sumar todos los tipos de comandos geográficos: geoadd, geodist, geohash, geopos, georadius y georadius bymember.	Recuento
GeoSpatialBasedCmdsLatency	Latencia de comandos basados en condiciones geoespaciales.	Microsegundos
GetTypeCmds	El número total de comandos de escritura de read-only. Se obtiene de la estadística de commandstats sumando todos los tipos de comandos read-only (get, hget, scard, lrange, etc.).	Recuento
GetTypeCmdsLatency	Latencia de los comandos de lectura.	Microsegundos
HashBasedCmds	El número total de comandos basados en hash. Se obtiene de la estadística de commandstats sumando todos los comandos que actúan en uno o más algoritmos hash (hget, hkeys, hvals, hdel, etc.).	Recuento
HashBasedCmdsLatency	Latencia de comandos basados en hash.	Microsegundos
HyperLogLogBasedCmds	El número total de comandos basados en HyperLogLog . Se obtiene de la estadística de commandstats sumando todos los tipos de comandos pf (pfadd, pfcount, pfmerge, etc.).	Recuento

Métrica	Descripción	Unidad
HyperLogLog BasedCmdsLatency	Latencia de los comandos basados en HyperLogLog.	Microsegundos
JsonBasedCmds	El número total de comandos JSON, incluidos los comandos de lectura y escritura. Se obtiene de la estadística de commandstats sumando todos los comandos de JSON que actúan sobre claves de JSON.	Recuento
JsonBasedCmdsLatency	Latencia de todos los comandos JSON, incluidos los comandos de lectura y escritura.	Microsegundos
JsonBasedGetCmds	Número total de comandos de solo lectura de JSON. Se obtiene de la estadística de commandstats sumando todos los comandos de lectura de JSON que actúan sobre claves de JSON.	Recuento
JsonBasedGetCmdsLatency	Latencia de los comandos de solo lectura de JSON.	Microsegundos
JsonBasedSetCmds	Número total de comandos de escritura de JSON. Se obtiene de la estadística de commandstats sumando todos los comandos de escritura de JSON que actúan sobre claves de JSON.	Recuento
JsonBasedSetCmdsLatency	Latencia de los comandos de escritura de JSON.	Microsegundos
KeyBasedCmds	El número total de comandos basados en claves. Se obtiene de la estadística de commandstats sumando todos los comandos que actúan en una o más claves en varias estructuras de datos (del, expire, rename, etc.).	Recuento

Métrica	Descripción	Unidad
KeyBasedCmdsLatency	Latencia de comandos basados en claves.	Microsegundos
ListBasedCmds	El número total de comandos basados en listas. Se obtiene de la estadística de commandstats sumando todos los comandos que actúan en una o más listas (lindex, lrange, lpush, ltrim, etc.).	Recuento
ListBasedCmdsLatency	Latencia de comandos basados en listas.	Microsegundos
NonKeyTypeCmds	Número total de comandos no basados en claves. Se obtiene de la estadística de commandstats sumando todos los comandos que no actúan sobre una clave, por ejemplo, acl, dbsize o info.	Recuento
NonKeyTypeCmdsLatency	Latencia de los comandos no basados en claves.	Microsegundos
PubSubBasedCmds	El número total de comandos para la funcionalidad publicación/suscripción. Se obtiene de la estadística de commandstats sumando todos los comandos utilizados para la funcionalidad de publicación/suscripción: psubscribe, publish, pubsub, punsubscribe, ssubscribe, sunsubscribe, spublish, subscribe y unsubscribe.	Recuento
PubSubBasedCmdsLatency	Latencia de comandos basados en publicación/suscripción.	Microsegundos

Métrica	Descripción	Unidad
SetBasedCmds	El número total de comandos basados en instrucciones set. Se obtiene de la estadísti ca de commandstats sumando todos los comandos que actúan en uno o más conjuntos (scard, sdiff, sadd, sunion, etc.).	Recuento
SetBasedCmd sLatency	Latencia de comandos basados en conjuntos.	Microsegu ndos
SetTypeCmds	El número total de tipos de comandos de write. Se obtiene de la estadística de commandst ats sumando todos los tipos de comandos mutative que actúan en los datos (set, hset. sadd, lpop, etc.).	Recuento
SetTypeCmdsLatency	Latencia de comandos de escritura.	Microsegu ndos
SortedSetBasedCmds	El número total de comandos basados en instrucciones set ordenadas. Se obtiene de la estadística de commandstats sumando todos los comandos que actúan en uno o más conjuntos ordenados (zcount, zrange, zrank, zadd, etc.).	Recuento
SortedSetBa sedCmdsLatency	Latencia de comandos basados en orden.	Microsegu ndos
StringBasedCmds	El número total de comandos basados en cadenas. Se obtiene de la estadística de commandstats sumando todos los comandos que actúan en una o más cadenas (strlen, setex, setrange, etc.).	Recuento
StringBased CmdsLatency	Latencia de comandos basados en cadenas	Microsegu ndos

Métrica	Descripción	Unidad
StreamBasedCmds	El número total de comandos basados en secuencias. Se obtiene de la estadística de commandstats sumando todos los comandos que actúan en uno o más tipos de datos de flujo (xrange, xlen, xadd, xdel, etc.).	Recuento
StreamBasedCmdsLatency	Latencia de comandos basados en secuencias.	Microsegundos
SearchBasedCmds	Número total de comandos Search, incluidos los comandos de lectura y escritura. Se obtiene de la estadística commandstats mediante la suma de todos los comandos Search.	Recuento
SearchBasedCmdsLatency	Latencia de todos los comandos Search, incluidos los comandos de lectura y escritura.	Microsegundos
SearchBasedGetCmds	Número total de comandos de solo lectura Search. Se obtiene de la estadística commandstats mediante la suma de todos los comandos de lectura Search.	Recuento
SearchBasedGetCmdsLatency	Latencia de los comandos de solo lectura Search.	Microsegundos
SearchBasedSetCmds	Número total de comandos de escritura Search. Se obtiene de la estadística commandstats mediante la suma de todos los comandos de escritura Search.	Recuento
SearchBasedSetCmdsLatency	Latencia de los comandos de escritura Search.	Microsegundos

Métricas de Memcached

El espacio de nombres de AWS/ElastiCache incluye las siguientes métricas Memcached.

El espacio de nombres de AWS/ElastiCache incluye las siguientes métricas, que se obtienen con el comando de estadísticas de Memcached. Cada métrica se calcula en el nivel de nodo de caché.

Consulte también

- Métricas de nivel de host

Métrica	Descripción	Unidad
`BytesReadIntoMemcached`	El número de bytes que el nodo de caché ha leído de la red.	Bytes
`BytesUsedForCacheItems`	El número de bytes usados para almacenar elementos de caché.	Bytes
`BytesWrittenOutFromMemcached`	El número de bytes que el nodo de caché ha escrito en la red.	Bytes
`CasBadval`	El número de solicitudes CAS (comprobar y establecer) que la caché ha recibido en las que el valor de Cas no coincidía con el valor de Cas almacenado.	Recuento
`CasHits`	El número de solicitudes Cas que la caché ha recibido en las que se encontró la clave solicitada y en las que el valor de Cas coincidía .	Recuento
`CasMisses`	El número de solicitudes Cas que la caché ha recibido en las que se no encontró la clave solicitada.	Recuento
`CmdFlush`	El número de comandos flush recibidos por la caché.	Recuento
`CmdGet`	El número de comandos get recibidos por la caché.	Recuento

Métrica	Descripción	Unidad
`CmdSet`	El número de comandos establecidos que ha recibido la caché.	Recuento
`CurrConnections`	Recuento del número de conexiones realizadas en la caché en un instante en el tiempo. ElastiCache usa dos de las tres conexiones para supervisar el clúster. Además de lo anterior, memcached crea un número de conexiones internas igual al doble de subprocesos utilizados para el tipo de nodo. El número de subprocesos para los distintos tipos de nodos se puede ver en los `Nodetype Specific Parameters` del grupo de parámetros aplicable. El total de conexiones es la suma de las conexiones de cliente, las conexiones para monitoreo y las conexiones internas mencionadas anteriormente.	Recuento
`CurrItems`	Recuento del número de elementos almacenados actualmente en la caché.	Recuento
`DecrHits`	El número de solicitudes decrement que la caché ha recibido en las que se encontró la clave solicitada.	Recuento
`DecrMisses`	El número de solicitudes decrement que la caché ha recibido en las que no se encontró la clave solicitada.	Recuento
`DeleteHits`	El número de solicitudes delete que la caché ha recibido en las que se encontró la clave solicitada.	Recuento

Métrica	Descripción	Unidad
DeleteMisses	El número de solicitudes delete que la caché ha recibido en las que no se encontró la clave solicitada.	Recuento
Evictions	El número de elementos no caducados que la caché desalojó para dejar espacio a nuevas operaciones de escritura.	Recuento
GetHits	El número de solicitudes get que la caché ha recibido en las que se encontró la clave solicitada.	Recuento
GetMisses	El número de solicitudes get que la caché ha recibido en las que no se encontró la clave solicitada.	Recuento
IncrHits	El número de solicitudes increment que la caché ha recibido en las que se encontró la clave solicitada.	Recuento
IncrMisses	El número de solicitudes increment que la caché ha recibido en las que no se encontró la clave solicitada.	Recuento
Reclaimed	El número de elementos caducados que la caché desalojó para dejar espacio a nuevas operaciones de escritura.	Recuento

Para Memcached 1.4.14, se proporcionan las siguientes métricas adicionales.

Métrica	Descripción	Unidad
BytesUsedForHash	El número de bytes usados actualmente por tablas hash.	Bytes
CmdConfigGet	El número acumulado de solicitudes config get.	Recuento

Métrica	Descripción	Unidad
CmdConfigSet	El número acumulado de solicitudes config set.	Recuento
CmdTouch	El número acumulado de solicitudes touch.	Recuento
CurrConfig	El número actual de configuraciones almacenadas.	Recuento
EvictedUnfetched	El número de elementos válidos desalojados de la caché usada menos recientemente (LRU) que no se manipularon nunca después de haber sido definidos.	Recuento
ExpiredUnfetched	El número de elementos caducados reclamados desde la LRU que no se tocaron nunca después de haber sido definidos.	Recuento
SlabsMoved	El número total de páginas slab que se han movido.	Recuento
TouchHits	El número de clases que se han manipulado y que recibieron una nueva hora de vencimiento.	Recuento
TouchMisses	El número de elementos que se han manipulado pero que no se encontraron.	Recuento

El espacio de nombres de AWS/ElastiCache incluye las siguientes métricas calculadas de nivel de caché.

Métrica	Descripción	Unidad
NewConnections	El número de conexiones nuevas recibidas por la caché. Este cálculo se obtiene de la estadística total_connections de Memcached registrando el cambio en total_connections durante un periodo de tiempo. Siempre será al menos 1,	Recuento

Métrica	Descripción	Unidad
	ya que hay una conexión reservada para un ElastiCache.	
`NewItems`	El número de elementos nuevos que se han almacenado. Este cálculo se obtiene de la estadística total_items de Memcached registrando el cambio en total_items durante un periodo de tiempo.	Recuento
`UnusedMemory`	La cantidad de memoria no usada por los datos. Este cálculo se obtiene de la estadística limit_maxbytes y bytes de Memcached sustrayendo bytes de limit_maxbytes. Como la carga de Memcached utiliza memoria además de la memoria que usan los datos, la memoria sin usar no debe considerarse la cantidad de memoria disponible para datos adicionales. Es posible que se desalojen algunos elementos aunque aún le quede memoria sin usar. Para obtener información más detallada, consulte Memcached item memory usage.	Bytes

¿Qué métricas debo monitorear?

Las siguientes métricas de CloudWatch ofrece buena información sobre el rendimiento de ElastiCache. En la mayoría de los casos, recomendamos que configure alarmas de CloudWatch para estas métricas con el fin de poder emprender acciones correctivas antes de que se produzca el problema de rendimiento.

Métricas que se van a monitorear

- CPUUtilization
- EngineCPUUtilization
- SwapUsage (Valkey y Redis OSS)
- Evictions
- CurrConnections
- Memoria (Valkey y Redis OSS)
- Network
- Latencia
- Replicación
- Administración del tráfico (Valkey y Redis OSS)

CPUUtilization

Se trata de una métrica de nivel de host que muestra un valor como un porcentaje. Para obtener más información, consulte Métricas de nivel de host.

Valkey y Redis OSS

En los tipos de nodos pequeños que tienen dos CPU virtuales o menos, utilice la métrica `CPUUtilization` para monitorear la carga de trabajo.

En general, sugerimos que establezca el umbral en el 90 % del ancho de banda de la CPU disponible. Debido a que Valkey y Redis OSS usan un único subproceso, el valor del umbral real se debe calcular como una fracción de la capacidad total del nodo. Por ejemplo, supongamos que está usando un tipo de nodo con dos núcleos. En este caso, el umbral de CPUUtilization sería de 90/2, es decir, el 45 %.

Deberá determinar su propio umbral en función del número de núcleos del nodo de caché que use. Si supera este umbral y su carga de trabajo principal es de solicitudes de lectura, escale el clúster de

forma ascendente agregando réplicas de lectura. Si la carga de trabajo principal es de solicitudes de escritura, en función de la configuración del clúster, recomendamos que:

- Clústeres de Valkey o Redis OSS (modo de clúster deshabilitado): escale verticalmente mediante un tipo de instancia con una caché de mayor tamaño.
- Clústeres de Valkey o Redis OSS (modo de clúster habilitado): añada más particiones para distribuir la carga de trabajo de escritura entre más nodos principales.

> **Tip**
>
> En lugar de utilizar la métrica de host `CPUUtilization`, los usuarios de Valkey y Redis OSS podrían utilizar la métrica `EngineCPUUtilization`, que informa del porcentaje de uso en el núcleo del motor de Valkey o Redis OSS. Para ver si esta métrica está disponible en sus nodos y para obtener más información, consulte Métricas de Valkey y Redis OSS.

Es posible que, en los tipos de nodos con cuatro o más vCPU, desee utilizar la métrica `EngineCPUUtilization`, que indica el porcentaje de uso del núcleo del motor de Valkey o Redis OSS. Para ver si esta métrica está disponible en sus nodos y para obtener más información, consulte Metrics for Redis OSS.

Memcached

Como Memcached usa múltiples subprocesos, esta métrica puede llegar hasta el 90 %. Si supera este umbral, escale su clúster verticalmente con un tipo de nodo de caché de mayor tamaño o escálelo horizontalmente añadiendo más nodos de caché.

EngineCPUUtilization

Es posible que, en los tipos de nodos más grandes, con cuatro o más vCPU, desee utilizar la métrica `EngineCPUUtilization`, que indica el porcentaje de uso del núcleo del motor de Redis OSS. Para ver si esta métrica está disponible en sus nodos y para obtener más información, consulte Métricas de Valkey y Redis OSS.

Para obtener más información, consulte la sección CPUs en Monitoring best practices with Amazon ElastiCache for Redis OSS using Amazon CloudWatch.

SwapUsage (Valkey y Redis OSS)

Se trata de una métrica de nivel de host que muestra un valor en bytes. Para obtener más información, consulte Métricas de nivel de host.

La métrica de CloudWatch `FreeableMemory` que está cerca de 0 (es decir, por debajo de 100 MB) o la métrica `SwapUsage` mayor que la métrica `FreeableMemory` indica que un nodo está bajo presión de memoria. Si esto sucede, consulte los siguientes temas:

- Forma de garantizar que dispone de memoria suficiente para crear una instantánea de Valkey o Redis OSS
- Administración de la memoria reservada para Valkey y Redis OSS

Evictions

Es una métrica del motor de la caché Recomendamos que determine su propio umbral de alarma para esta métrica en función de las necesidades de su aplicación.

Si está utilizando Memcached y supera el umbral elegido, escale verticalmente su clúster utilizando un tipo de nodo mayor o escálelo horizontalmente añadiendo más nodos.

CurrConnections

Es una métrica del motor de la caché Recomendamos que determine su propio umbral de alarma para esta métrica en función de las necesidades de su aplicación.

Un número de CurrConnections cada vez mayor puede indicar que existe un problema con la aplicación. Por lo tanto, deberá investigar el comportamiento de la aplicación antes de abordar este problema.

Para obtener más información, consulte la sección Connections en Monitoring best practices with Amazon ElastiCache for Redis OSS using Amazon CloudWatch.

Memoria (Valkey y Redis OSS)

La memoria es un aspecto central de Valkey y Redis OSS. Es necesario comprender la utilización de la memoria de un clúster para evitar la pérdida de datos y adaptarse al crecimiento futuro del conjunto de datos. Las estadísticas sobre la utilización de memoria de un nodo se encuentran disponibles en la sección de memoria del comando INFO.

Para obtener más información, consulte la sección Memory en Monitoring best practices with Amazon ElastiCache for Redis OSS using Amazon CloudWatch.

Network

Uno de los factores determinantes de la capacidad de la banda ancha de red del clúster es el tipo de nodo seleccionado. Para obtener más información sobre la capacidad de red del nodo, consulte Precios de Amazon ElastiCache.

Para obtener más información, consulte la sección Network en Monitoring best practices with Amazon ElastiCache for Redis OSS using Amazon CloudWatch.

Latencia

La medición del tiempo de respuesta de una instancia de ElastiCache para Valkey se puede abordar de varias maneras en función del nivel de detalle requerido. Las etapas clave que contribuyen al tiempo de respuesta general del servidor de ElastiCache para Valkey son el procesamiento previo, la ejecución y el procesamiento posterior de comandos.

Las métricas de latencia específicas de los comandos derivadas del comando INFO de Valkey, como GetTypeCmdsLatency y SetTypeCmdsLatency, se centran específicamente en ejecutar la lógica de comandos básica del comando Valkey. Estas métricas le resultarán útiles si lo que desea es determinar el tiempo de ejecución del comando o las latencias agregadas por estructura de datos.

Las métricas de latencia `SuccessfulWriteRequestLatency` y `SuccessfulReadRequestLatency` miden el tiempo total que tarda el motor de ElastiCache para Valkey en responder a una solicitud.

> **Note**
>
> Cuando se utiliza la canalización de Valkey con CLIENT REPLY habilitado en el cliente de Valkey, puede que los valores para las métricas `SuccessfulWriteRequestLatency` y `SuccessfulReadRequestLatency` se inflen. La canalización de Valkey es una técnica para mejorar el rendimiento mediante la emisión de varios comandos a la vez, sin esperar a que se responda a cada comando individual. Para evitar que se inflen los valores, recomendamos configurar el cliente Valkey para que canalice los comandos con CLIENT REPLY OFF.

Para obtener más información, consulte la sección Latency en Monitoring best practices with Amazon ElastiCache using Amazon CloudWatch.

Replicación

El volumen de datos que se replican es visible a través de la métrica `ReplicationBytes`. Aunque esta métrica representa la carga de escritura del grupo de replicación, no proporciona información acerca del estado de replicación. Para este propósito, puede utilizar la métrica `ReplicationLag`.

Para obtener más información, consulte la sección Replication en Monitoring best practices with Amazon ElastiCache para Redis OSS using Amazon CloudWatch.

Administración del tráfico (Valkey y Redis OSS)

ElastiCache para Redis OSS administra automáticamente el tráfico de un nodo cuando se envían al nodo más comandos de los que Valkey o Redis OSS pueden procesar. Esto se hace para mantener un funcionamiento y una estabilidad óptimos del motor.

Cuando el tráfico se administra activamente en un nodo, la métrica `TrafficManagementActive` emite puntos de datos de 1. Esto indica que el nodo ha disminuido la escala para la carga de trabajo que se proporciona. Si esta métrica sigue siendo 1 durante largos periodos de tiempo, evalúe el clúster para decidir si es necesario escalar verticalmente o escalar horizontalmente.

Para obtener más información, consulte la métrica `TrafficManagementActive` en la página Métricas.

Elección de periodos y estadísticas de métricas

Aunque CloudWatch permitirá elegir cualquier periodo y estadística para cada métrica, tenga en cuenta que no todas las combinaciones serán útiles. Por ejemplo, las estadísticas Average, Minimum y Maximum son útiles para CPUUtilization; sin embargo, la estadística Sum no lo es.

Todas las muestras de ElastiCache se publican por un periodo de 60 segundos para cada nodo de caché individual. La métrica de nodo de caché solo contendrá una única muestra para cualquier periodo de 60 segundos.

Para obtener más información acerca de cómo recuperar las métricas para los nodos de caché, consulte Monitoreo de métricas de nodo y clúster de CloudWatch.

Monitoreo de métricas de nodo y clúster de CloudWatch

ElastiCache y CloudWatch se encuentran integrados para ofrecer una amplia variedad de métricas. Estas métricas se pueden monitorear mediante CloudWatch.

> **Note**
>
> El siguiente ejemplo requiere el uso de las herramientas de línea de comandos de CloudWatch. A fin de obtener más información sobre CloudWatch y cómo descargar las herramientas para desarrolladores, consulte la página de producto de CloudWatch.

Los siguientes procedimientos muestran cómo utilizar CloudWatch a fin de recopilar estadísticas de espacio de almacenamiento para un clúster en la última hora.

> **Note**
>
> Los valores de `StartTime` y `EndTime` proporcionados en los ejemplos siguientes se proporcionan con fines ilustrativos. Deberá sustituir los valores de hora de inicio y finalización para sus nodos de caché.

Para obtener más información sobre los límites de ElastiCache, consulte AWS Service Limits de ElastiCache.

Monitoreo de métricas de nodo y clúster de CloudWatch (consola)

Para recopilar estadísticas de uso de la CPU de un clúster de caché

1. Inicie sesión en la Consola de administración de AWS y abra la consola de ElastiCache en https://console.aws.amazon.com/elasticache/.
2. Seleccione los nodos de caché de los que desea ver métricas.

   > **Note**
   >
   > La selección de más de 20 nodos deshabilita la visualización de métricas en la consola.

   a. En la página Clústeres de caché de la Consola de administración de AWS, haga clic en el nombre de uno o varios clústeres.

      Aparecerá la página de detalles del clúster.

   b. Haga clic en la pestaña Nodos situada en la parte superior de la ventana.

   c. En la pestaña Nodes de la ventana de detalles, seleccione los nodos de caché para los que desea ver métricas.

      Aparecerá una lista de las métricas de CloudWatch en la parte inferior de la ventana de la consola.

   d. Haga clic en la métrica CPU Utilization.

      Se abrirá la consola de CloudWatch y se mostrarán las métricas seleccionadas. Puede usar los cuadros de lista desplegables Statistic y Period y la pestaña Time Range para cambiar las métricas mostradas.

Monitoreo de las métricas de nodo y clúster de CloudWatch mediante la CLI de CloudWatch

Para recopilar estadísticas de uso de la CPU de un clúster de caché

- Para Linux, macOS o Unix:

```
aws cloudwatch get-metric-statistics \
    --namespace AWS/ElastiCache \
```

```
    --metric-name CPUUtilization \
    --dimensions='[{"Name":"CacheClusterId","Value":"test"},
{"Name":"CacheNodeId","Value":"0001"}]' \
    --statistics=Average \
    --start-time 2018-07-05T00:00:00 \
    --end-time 2018-07-06T00:00:00 \
    --period=3600
```

Para Windows:

```
aws cloudwatch get-metric-statistics ^
    --namespace AWS/ElastiCache ^
    --metric-name CPUUtilization ^
    --dimensions='[{"Name":"CacheClusterId","Value":"test"},
{"Name":"CacheNodeId","Value":"0001"}]' ^
    --statistics=Average ^
    --start-time 2018-07-05T00:00:00 ^
    --end-time 2018-07-06T00:00:00 ^
    --period=3600
```

Monitoreo de las métricas de clúster y nodo de CloudWatch mediante la API de CloudWatch

Para recopilar estadísticas de uso de la CPU de un clúster de caché

- Llame a la API de CloudWatch `GetMetricStatistics` con los siguientes parámetros (tenga en cuenta que las horas de inicio y finalización solo se muestran como ejemplos; tendrá que sustituir estos valores por sus propias horas de inicio y finalización):
  - `Statistics.member.1=Average`
  - `Namespace=AWS/ElastiCache`
  - `StartTime=2013-07-05T00:00:00`
  - `EndTime=2013-07-06T00:00:00`
  - `Period=60`
  - `MeasureName=CPUUtilization`
  - `Dimensions=CacheClusterId=mycachecluster,CacheNodeId=0002`

Example

```
http://monitoring.amazonaws.com/
    ?Action=GetMetricStatistics
    &SignatureVersion=4
    &Version=2014-12-01
    &StartTime=2018-07-05T00:00:00
    &EndTime=2018-07-06T23:59:00
    &Period=3600
    &Statistics.member.1=Average
    &Dimensions.member.1="CacheClusterId=mycachecluster"
    &Dimensions.member.2="CacheNodeId=0002"
    &Namespace=&AWS;/ElastiCache
    &MeasureName=CPUUtilization
    &Timestamp=2018-07-07T17%3A48%3A21.746Z
    &AWS;AccessKeyId=<&AWS; Access Key ID>
    &Signature=<Signature>
```

Cuotas de ElastiCache

La cuenta de AWS tiene cuotas predeterminadas para cada servicio de AWS (estas cuotas anteriormente se denominaban “límites”). A menos que se indique lo contrario, cada cuota es específica de la región de . Puede solicitar el aumento de algunas cuotas, pero otras no se pueden aumentar.

Para ver las cuotas de ElastiCache, abra la consola de Service Quotas. En el panel de navegación, elija Servicios de AWS y seleccione ElastiCache.

Para solicitar un aumento de cuota, consulte Solicitud de aumento de cuota en la Guía del usuario de Service Quotas. Si la cuota aún no se encuentra disponible en Service Quotas, utilice el formulario de aumento del límite.

La cuenta de AWS incluye las siguientes cuotas en relación con ElastiCache.

Nombre	Valor predeterminado	Descripción	Nombre de métrica
Nodos por región	300	Número máximo de nodos en todos los clústeres de una región. Esta cuota se aplica a los nodos no reservados de la región especific ada. Puede tener hasta 300 nodos no reservados en la misma región.	NodesPerRegion
Nodos por clúster (Memcached)	60	Número máximo de nodos de un clúster de Memcached individual.	Memcached NodesPerCluster
Nodos por clúster (modo de clúster habilitado)	90	Número máximo de nodos de un clúster	NodesPerCluster

Nombre	Valor predeterminado	Descripción	Nombre de métrica
		de Redis OSS o Valkey individual.	
Grupos de parámetros por región	300	Número máximo de grupos de parámetros que se pueden crear en una región.	ParameterGroup
Grupos de subredes por región	300	Número máximo de grupos de subredes que se pueden crear en una región.	SubnetGroup
Subredes por grupo de subredes	20	Número máximo de subredes que se pueden definir para un grupo de subredes.	SubnetsPerSubnetGroup
Cachés sin servidor por región	40	Número máximo de cachés sin servidor en una región.	ServerlessCache
Instantáneas sin servidor por día y por caché	24	Número máximo de instantáneas sin servidor que se pueden tomar por día por caché sin servidor.	SnapshotsPerDayPerCache
Usuarios por región	2000	Número total máximo de usuarios que se pueden crear en una región.	Usuario

Nombre	Valor predeterminado	Descripción	Nombre de métrica
Grupos de usuarios por región	200	Número máximo de grupos de usuarios que se pueden crear en una región.	UserGroup
Usuarios por grupo de usuarios	100	Número máximo de usuarios que se pueden definir para un grupo de usuarios.	UsersPerUserGroup

Referencia

En los temas de esta sección se abordan el trabajo con la API de Amazon ElastiCache y la sección ElastiCache de la AWS CLI. También se describen mensajes de error y notificaciones de servicio comunes.

- Uso de la API de ElastiCache
- Referencia de la API de ElastiCache
- Sección ElastiCache de la referencia de la AWS CLI
- Mensajes de error de Amazon ElastiCache
- Notificaciones

Uso de la API de ElastiCache

Esta sección proporciona descripciones orientadas a tareas sobre el uso y la implementación de operaciones de ElastiCache. Para obtener una descripción completa de dichas operaciones, consulte la Referencia de la API de Amazon ElastiCache.

Temas

- Uso de la API de consultas
- Bibliotecas disponibles
- Solución de problemas de aplicaciones

Uso de la API de consultas

Parámetros de consulta

Las solicitudes basadas en consultas HTTP son solicitudes HTTP que utilizan el verbo HTTP GET o POST y un parámetro de consulta denominado `Action`.

Cada solicitud de consulta debe incluir algunos parámetros comunes para realizar la autenticación y la selección de una acción.

Algunas operaciones toman listas de parámetros. Estas listas se especifican utilizando la notación `param.`*n*. Los valores de *n* son números enteros a partir de 1.

Autenticación de solicitudes de consulta

Solo se pueden enviar solicitudes de consulta a través de HTTPS y cada una de ellas debe incluir una firma. En esta sección se describe cómo crear la firma. El método que se describe en el procedimiento siguiente se conoce como firma versión 4.

A continuación se indican los pasos básicos que se utilizan para autenticar las solicitudes en AWS. En este proceso se presupone que se ha registrado en AWS y que dispone de un ID de clave de acceso y una clave de acceso secreta.

Proceso de autenticación de consulta

1. El remitente crea una solicitud para AWS.
2. El remitente calcula la firma de la solicitud, una operación hash para código de autenticación de mensajes (HMAC) basado en hash mediante una función hash SHA-1, tal y como se define en la siguiente sección de este tema.
3. El remitente de la solicitud envía a los datos de la misma, la firma y el ID de clave de acceso (el identificador de clave de la clave de acceso secreta utilizada) AWS.
4. AWS utiliza el ID de clave de acceso para buscar la clave de acceso secreta.
5. AWS genera una firma a partir de los datos de la solicitud y la clave de acceso secreta con el mismo algoritmo que se utilizó para calcular la firma de la solicitud.
6. Si las firmas coinciden, se considera que la solicitud es auténtica. Si la comparación falla, se descarta la solicitud y AWS devuelve una respuesta de error.

> **Note**
>
> Si una solicitud contiene un parámetro `Timestamp`, la firma calculada para la solicitud caduca 15 minutos después de su valor.
> Si una solicitud contiene un parámetro `Expires`, la firma caduca en el momento especificado por el parámetro `Expires`.

Para calcular la firma de la solicitud

1. Cree la cadena de consulta canónica que necesitará más adelante en este procedimiento:

a. Ordene los componentes UTF-8 de la cadena de consulta por nombre de parámetro con el orden de bytes natural. Los parámetros pueden proceder de GET URI o del cuerpo de la solicitud POST (cuando Content-Type es application/x-www-form-urlencoded).

b. Codifique como dirección URL el nombre y los valores del parámetro, aplicando las reglas siguientes:

   i. No incluya en la codificación de la dirección URL ninguno de los caracteres no reservados definidos en la norma RFC 3986. Estos caracteres no reservados son A–Z, a–z, 0–9, guion (-), carácter de subrayado (_), punto (.) y tilde (~).

   ii. Codifique con signos de porcentaje el resto de los caracteres con %XY, donde X e Y son caracteres hexadecimales (0-9 y A-F mayúsculas).

   iii. Codifique con signos de porcentaje los caracteres extendidos UTF-8 con el formato %XY%ZA...

   iv. Codifique con el signo de porcentaje el carácter de espacio como %20 y no como + (lo que se hace en las codificaciones comunes).

c. Separe los nombres de los parámetros codificados de sus valores codificados con el signo igual (=) (carácter ASCII 61), aunque el valor del parámetro esté vacío.

d. Separe los pares de nombre-valor con el carácter ampersand (&) (código ASCII 38).

2. Cree la cadena para firmar según la siguiente pseudogramática ("\n" representa un carácter de nueva línea ASCII).

```
StringToSign = HTTPVerb + "\n" +
ValueOfHostHeaderInLowercase + "\n" +
HTTPRequestURI + "\n" +
CanonicalizedQueryString <from the preceding step>
```

   El componente HTTPRequestURI es el componente de la ruta absoluta HTTP del URI hasta la cadena de consulta, pero sin incluirla. Si HTTPRequestURI está vacío, utilice una barra diagonal (/).

3. Calcule una HMAC conforme con RFC 2104 con la cadena que acaba de crear, su clave de acceso secreta como la clave y SHA256 o SHA1 como algoritmo de hash.

   Para obtener más información, consulte https://www.ietf.org/rfc/rfc2104.txt.

4. Convierta el valor resultante en base 64.

5. Incluya el valor como valor del parámetro `Signature` de la solicitud.

A continuación se muestra una solicitud de muestra (se han agregado saltos de línea para facilitar la lectura).

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=DescribeCacheClusters
    &CacheClusterIdentifier=myCacheCluster
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &Version=2014-12-01
```

Para la cadena de consulta anterior, debería calcular la firma HMAC de la siguiente cadena.

```
GET\n
    elasticache.amazonaws.com\n
    Action=DescribeCacheClusters
    &CacheClusterIdentifier=myCacheCluster
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &Version=2014-12-01
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Credential=AKIADQKE4SARGYLE%2F20140523%2Fus-west-2%2Felasticache
%2Faws4_request
    &X-Amz-Date=20141201T223649Z
    &X-Amz-SignedHeaders=content-type%3Bhost%3Buser-agent%3Bx-amz-content-sha256%3Bx-
amz-date
        content-type:
        host:elasticache.us-west-2.amazonaws.com
        user-agent:CacheServicesAPICommand_Client
    x-amz-content-sha256:
    x-amz-date:
```

El resultado es la siguiente solicitud firmada.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=DescribeCacheClusters
    &CacheClusterIdentifier=myCacheCluster
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &Version=2014-12-01
```

```
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Credential=AKIADQKE4SARGYLE/20141201/us-west-2/elasticache/aws4_request
    &X-Amz-Date=20141201T223649Z
    &X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
    &X-Amz-Signature=2877960fced9040b41b4feaca835fd5cfeb9264f768e6a0236c9143f915ffa56
```

Para obtener información detallada acerca del proceso de firma y el cálculo de la firma de la solicitud, consulte el tema Proceso de firma Signature Version 4 y sus subtemas.

Bibliotecas disponibles

AWS ofrece kits de desarrollo de software (SDK) para los desarrolladores de software que prefieran crear aplicaciones mediante las API de lenguaje específico en lugar de la API de consulta. Estos SDK proporcionan funciones básicas (que no se incluyen en las API), como la autenticación de solicitudes, los reintentos de solicitudes y la gestión de errores para que se pueda comenzar más fácilmente. Encontrará SDK y recursos adicionales para los siguientes lenguajes de programación:

- Java
- Windows y .NET
- PHP
- Python
- Ruby

Para obtener información acerca de otros lenguajes, consulte las Código de muestra y bibliotecas.

Solución de problemas de aplicaciones

ElastiCache proporciona errores específicos y descriptivos para ayudarlo a solucionar problemas durante la interacción con la API de ElastiCache.

Recuperación de errores

Normalmente, conviene que una aplicación compruebe si una solicitud generó un error antes de emplear tiempo en procesar los resultados. La forma más fácil de averiguar si se ha producido un error, consiste en buscar un nodo de `Error` en la respuesta de la API de ElastiCache.

La sintaxis XPath permite comprobar fácilmente si hay un nodo `Error`, y ofrece un método sencillo de recuperar el mensaje de error y su código. La partición de código siguiente utiliza Perl y el módulo

XML::XPath para determinar si se ha producido un error durante una solicitud. Si es así, el código imprime el primer mensaje de error y su código en la respuesta.

```
use XML::XPath;
my $xp = XML::XPath->new(xml =>$response);
if ( $xp->find("//Error") )
{print "There was an error processing your request:\n", " Error code: ",
$xp->findvalue("//Error[1]/Code"), "\n", " ",
$xp->findvalue("//Error[1]/Message"), "\n\n"; }
```

Consejos para la solución de problemas

Recomendamos los siguientes procesos para diagnosticar y solucionar problemas con la API de ElastiCache.

- Verifique si ElastiCache se ejecuta correctamente.

  Para ello, solo tiene que abrir una ventana del navegador y enviar una solicitud de consulta al servicio de ElastiCache (como https://elasticache.amazonaws.com). El mensaje 500 Error de servidor interno o MissingAuthenticationTokenException confirmará que el servicio está disponible y que responde a las solicitudes.
- Comprobar la estructura de la solicitud.

  Cada operación de ElastiCache tiene una página de referencia en la Referencia de la API de ElastiCache. Compruebe que utiliza los parámetros correctamente. Para obtener ideas sobre lo que podría estar mal, examine las solicitudes de muestra o los escenarios de usuario para ver si esos ejemplos realizan operaciones similares.
- Visite el foro.

  Existe un foro de discusión de ElastiCache donde puede buscar soluciones a los problemas que otras personas han experimentado al utilizar este servicio. Para ver el foro, consulte

  https://forums.aws.amazon.com/

Configuración de la interfaz de línea de comandos de ElastiCache

En esta sección, se describen los requisitos previos para ejecutar las herramientas de línea de comandos, se explica dónde se pueden obtener y cómo configurar las herramientas y su entorno, y se incluyen una serie de ejemplos sobre los usos habituales de las herramientas.

Siga las instrucciones de este tema únicamente si va a utilizar la AWS CLI de ElastiCache.

> **Important**
>
> La interfaz de línea de comandos (CLI) de Amazon ElastiCache no admite mejoras de ElastiCache después de la versión de la API del 30/09/2014. Para utilizar la funcionalidad de ElastiCache más reciente desde la línea de comandos, utilice la interfaz de la línea de comandos de AWS.

Temas

- Requisitos previos
- Obtención de las herramientas de línea de comandos
- Configuración de las herramientas
- Suministro de credenciales para las herramientas
- Variables de entorno

Requisitos previos

En este documento, se presupone que se trabaja en un entorno de Linux/UNIX o Windows. Las herramientas de línea de comandos de Amazon ElastiCache también funcionan en Mac OS X, que es un entorno basado en UNIX; sin embargo, en esta guía no se incluyen instrucciones específicas para Mac OS X.

Por convención, todo el texto de las líneas de comando va precedido de un símbolo de línea de comando genérico: **`PROMPT>`** . Es probable que el símbolo de línea de comando real que aparezca en su equipo sea distinto. También utilizamos **`$`** para representar comandos específicos de Linux/UNIX y **`C:\>`** para representar comandos específicos de Windows. La salida de ejemplo que se obtiene como resultado del comando se muestra inmediatamente después sin ningún prefijo.

El entorno en tiempo de ejecución de Java

Las herramientas de línea de comandos que se utilizan en esta guía requieren que se ejecute Java versión 5 o posterior. Se acepta una instalación de JRE o de JDK. Para ver y descargar los JRE para las distintas plataformas, como Linux/UNIX y Windows, consulte Descargas de Java SE.

Configuración de la variable Java Home

Las herramientas de línea de comandos dependen de una variable de entorno (`JAVA_HOME`) para localizar Java Runtime. Esta variable de entorno se debe establecer en la ruta completa del directorio que contiene un subdirectorio denominado `bin` que, a su vez, contiene el ejecutable `java` (en Linux y UNIX) o `java.exe` (en Windows).

Para establecer la variable Java_Home

1. Establezca la variable Java_Home.

   - En Linux y UNIX, escriba el comando siguiente:

     ```
     $ export JAVA_HOME=<PATH>
     ```

   - En Windows, escriba el siguiente comando:

     ```
     C:\> set JAVA_HOME=<PATH>
     ```

2. Confirme la configuración de la ruta ejecutando **`$JAVA_HOME/bin/java -version`** y comprobando la salida.

   - En Linux/UNIX, verá una salida similar a la siguiente:

     ```
     $ $JAVA_HOME/bin/java -version
     java version "1.6.0_23"
     Java(TM) SE Runtime Environment (build 1.6.0_23-b05)
     Java HotSpot(TM) Client VM (build 19.0-b09, mixed mode, sharing)
     ```

   - En Windows, verá una salida similar a la siguiente:

     ```
     C:\> %JAVA_HOME%\bin\java -version
     java version "1.6.0_23"
     Java(TM) SE Runtime Environment (build 1.6.0_23-b05)
     Java HotSpot(TM) Client VM (build 19.0-b09, mixed mode, sharing)
     ```

Obtención de las herramientas de línea de comandos

Las herramientas de línea de comandos se encuentran disponibles como archivo ZIP en el sitio web de herramientas para desarrolladores de ElastiCache. Estas herramientas están escritas en Java, e incluyen scripts de shell para Windows 2000/XP/Vista/Windows 7, Linux/UNIX y Mac OSX. El archivo ZIP es autónomo y no requiere instalación; basta con descargarlo y descomprimirlo en un directorio del equipo local.

Configuración de las herramientas

Las herramientas de línea de comandos dependen de una variable de entorno (AWS_ELASTICACHE_HOME) para localizar las bibliotecas que necesitan. Es necesario configurar esta variable de entorno para poder utilizar las herramientas. Establézcala en la ruta del directorio en el que descomprimió las herramientas de línea de comandos. Este directorio se denomina ElastiCacheCli-A.B.nnnn (A, B y n son números de versión) y contiene subdirectorios denominados bin y lib.

Para configurar la variable de entorno AWS_ELASTICACHE_HOME

- Abra una ventana de línea de comandos e ingrese uno de los siguientes comandos para establecer la variable de entorno AWS_ELASTICACHE_HOME.
  - En Linux y UNIX, escriba el comando siguiente:

```
$ export &AWS;_ELASTICACHE_HOME=<path-to-tools>
```

  - En Windows, escriba el siguiente comando:

```
C:\> set &AWS;_ELASTICACHE_HOME=<path-to-tools>
```

Para facilitar el uso de las herramientas, recomendamos que agregue el directorio BIN de herramientas a la ruta del sistema. En esta guía se presupone que el directorio BIN está en la ruta del sistema.

Para agregar el directorio BIN de las herramientas a la ruta del sistema

- Introduzca los siguientes comandos para agregar el directorio BIN de las herramientas a la ruta del sistema.

- En Linux y UNIX, escriba el comando siguiente:

```
$ export PATH=$PATH:$&AWS;_ELASTICACHE_HOME/bin
```

- En Windows, escriba el siguiente comando:

```
C:\> set PATH=%PATH%;%&AWS;_ELASTICACHE_HOME%\bin
```

> **Note**
>
> Las variables de entorno de Windows se restablecen cuando se cierra la ventana de comandos. Es posible que desee configurarlas de forma permanente. Para obtener más información, consulte la documentación de su versión de Windows.

> **Note**
>
> Las rutas que contienen espacios deben encerrarse entre comillas; por ejemplo: "C:\Program Files\Java"

Suministro de credenciales para las herramientas

Las herramientas de línea de comandos necesitan la clave de acceso y la clave de acceso secreta de AWS que se proporcionan con la cuenta de AWS. Puede obtenerlas desde la línea de comandos o desde un archivo de credenciales situado en el sistema local.

La implementación incluye un archivo de plantilla ${AWS_ELASTICACHE_HOME}/credential-file-path.template que se debe editar con información personalizada. A continuación se muestra el contenido del archivo de plantilla:

```
AWSAccessKeyId=<Write your AWS access ID>
AWSSecretKey=<Write your AWS secret key>
```

> **Important**
>
> En UNIX, limite los permisos al propietario del archivo de credenciales:
>
> ```
> $ chmod 600 <the file created above>
> ```

Una vez que el archivo de credenciales se encuentre configurado, tendrá que configurar la variable de entorno AWS_CREDENTIAL_FILE para que las herramientas de ElastiCache puedan encontrar la información.

Para configurar la variable de entorno AWS_CREDENTIAL_FILE

1. Establezca la variable de entorno:
   - En Linux y UNIX, actualice la variable con el comando siguiente:

     ```
     $ export &AWS;_CREDENTIAL_FILE=<the file created above>
     ```
   - En Windows, establezca la variable con el comando siguiente:

     ```
     C:\> set &AWS;_CREDENTIAL_FILE=<the file created above>
     ```
2. Compruebe que la configuración funciona correctamente; para ello, ejecute el comando siguiente:

   ```
   elasticache --help
   ```

   Debería ver la página de uso de todos los comandos de ElastiCache.

Variables de entorno

Las variables de entorno pueden resultar útiles para crear scripts, configurar valores predeterminados o anularlos temporalmente.

Además de la variable de entorno AWS_CREDENTIAL_FILE, la mayoría de las herramientas de la API incluidas en la interfaz de línea de comandos de ElastiCache admiten las siguientes variables:

- EC2_REGION: la región de AWS que se utilizará.

- AWS_ELASTICACHE_URL: la URL que se utilizará para la llamada de servicio. No es necesario especificar otro punto de enlace regional si se especifica EC2_REGION o se pasa el parámetro --region.

En los siguientes ejemplos se muestra cómo establecer la variable de entorno EC2_REGION para configurar la región utilizada por las herramientas de la API:

Linux, OS X o Unix

```
$ export EC2_REGION=us-west-1
```

Windows

```
$ set EC2_REGION=us-west-1
```

Mensajes de error de Amazon ElastiCache

Amazon ElastiCache devuelve los siguientes mensajes de error. Puede recibir otros mensajes de error devueltos por ElastiCache, otros servicios de AWS o Valkey, Redis OSS o Memcached. Para obtener descripciones de los mensajes de error de otros orígenes distintos a ElastiCache, consulte la documentación de la fuente que genera el mensaje de error.

- Cluster node quota exceeded
- Customer's node quota exceeded
- Manual snapshot quota exceeded
- Insufficient cluster capacity

Mensaje de error: se ha superado la cuota de nodos del clúster. Cada clúster puede tener como máximo %n nodos en esta región.

Causa: ha intentado crear o modificar un clúster y el resultado de dicha modificación daría lugar a un clúster con más de %n nodos.

Solución: cambie la solicitud para que el clúster no tenga más de %n nodos. O bien, si necesita más de %n nodos, realice la solicitud con el formulario de solicitud de nodos de Amazon ElastiCache.

Para obtener más información, consulte Límites de Amazon ElastiCache en Referencia general de Amazon Web Services.

Mensajes de error: se ha superado la cuota de nodos de clientes. Puede tener como máximo %n nodos en esta región O Ya ha alcanzado su cuota de %s nodos en esta región.

Causa: ha intentado crear o modificar un clúster y el resultado de dicha modificación es que su cuenta tendría más de %n nodos en todos los clústeres de esta región.

Solución: cambie la solicitud de forma que el total de nodos en la región de todos los clústeres para esta cuenta no supere %n nodos. O bien, si necesita más de %n nodos, realice la solicitud con el formulario de solicitud de nodos de Amazon ElastiCache.

Para obtener más información, consulte Límites de Amazon ElastiCache en Referencia general de Amazon Web Services.

Mensajes de error: The maximum number of manual snapshots for this cluster taken within 24 hours has been reached or The maximum number of manual snapshots for this node taken within 24 hours has been reached its quota of %n

Causa: ha intentado tomar una instantánea manual de un clúster cuando ya ha tomado el máximo permitido de instantáneas manuales en un periodo de 24 horas.

Solución: espere 24 horas para intentar tomar otra instantánea manual del clúster. O bien, si necesita tomar una instantánea manual ahora, tome la instantánea de otro nodo que tenga los mismos datos como, por ejemplo, un nodo diferente de un clúster.

Mensajes de error: InsufficientCacheClusterCapacity

Causa: AWS no tiene suficiente capacidad bajo demanda disponible para brindar servicio a la solicitud.

Solución:

- Espere unos minutos y después envíe la solicitud de nuevo; la capacidad puede cambiar frecuentemente.

- Envíe una nueva solicitud con una cantidad reducida de nodos o fragmentos (grupos de nodos). Por ejemplo, si hace una única solicitud para lanzar 15 nodos, intente hacer 3 solicitudes para 5 nodos o 15 solicitudes de 1 nodo en su lugar.
- Si está lanzando un clúster, envíe una solicitud nueva sin especificar ninguna zona de disponibilidad.
- Si está lanzando un clúster, envíe una solicitud nueva usando un tipo de nodo distinto (que puede ampliar en una fase posterior). Para obtener más información, consulte Escalado de ElastiCache.

Notificaciones

En este tema se abordan las notificaciones de ElastiCache que podrían interesarle. Una notificación es una situación o un evento que, en la mayoría de los casos, es temporal y solo se mantiene hasta que se encuentra y se implementa una solución. Las notificaciones suelen tener una fecha de inicio y una fecha de resolución. Pasada esta fecha, la notificación deja de ser relevante. Las notificaciones pueden ser relevantes o irrelevantes para usted. Las directrices de implementación que recomendamos permiten mejorar el desempeño de su clúster, siempre que decida seguirlas.

Las notificaciones no anuncian características o funcionalidad de ElastiCache nuevas o mejoradas.

Notificaciones generales de ElastiCache

Actualmente, no hay notificaciones destacables de ElastiCache que no sean específicas del motor.

Notificaciones de ElastiCache for Memcached

Las siguientes notificaciones de ElastiCache son específicas del motor de Memcached.

Notificaciones específicas de ElastiCache for Memcached

- Alerta: errores de segmentación provocados por Memcached LRU Crawler

Alerta: errores de segmentación provocados por Memcached LRU Crawler

⚠ Fecha de la alerta: 28 de febrero de 2017

En determinadas circunstancias, es posible que su clúster muestre cierta inestabilidad debido a un error de segmentación en LRU Memcached Crawler. Se trata de un problema en el motor de Memcached que ha estado presente durante un tiempo. El problema apareció en Memcached 1.4.33 cuando LRU Crawler se habilitaba de forma predeterminada.
Si tiene este problema, le recomendamos que deshabilite LRU Crawler hasta que haya una solución. Para ello, use `lru_crawler disable` en la línea de comandos o modifique el valor del parámetro `lru_crawler` (opción preferida).
Fecha de resolución:
Resolución:

Notificaciones específicas de ElastiCache para Redis OSS

Actualmente, no hay notificaciones destacables de ElastiCache para Redis OSS.

Historial de la documentación de ElastiCache

- Versión de la API: 2015-02-02
- Última actualización de la documentación: 27 de noviembre de 2023

En la siguiente tabla, se describen los cambios importantes en cada versión de la Guía del usuario de ElastiCache posteriores a marzo de 2018. Para obtener notificaciones sobre las actualizaciones de esta documentación, puede suscribirse a la fuente RSS.

Actualizaciones recientes de ElastiCache

Cambio	Descripción	Fecha
Compatibilidad con ElastiCache para el escalado vertical bajo demanda y el escalado horizontal automatizado de Memcached	ElastiCache con Memcached ahora admite el escalado vertical bajo demanda y el escalado horizontal automatizado. Esto incluye una actualización de la política de la acción `elasticache:ModifyCacheCluster` .	10 de abril de 2025
Compatibilidad con ElastiCache con Valkey	ElastiCache ahora es compatible con Valkey. Valkey 7.2.6 es compatible con Redis OSS 7.2. Para obtener más información, consulte Valkey.	8 de octubre de 2024
Nodos reservados de tamaño flexible	ElastiCache ahora admite nodos reservados de tamaño flexible. Para obtener más información, consulte Precios de Amazon ElastiCache.	1 de octubre de 2024

Soporte para ElastiCache para Memcached 1.6.22	ElastiCache para Memcached es ahora compatible con la versión 1.6.22 de Memcached . Incluye correcciones de errores acumulativas desde la versión 1.6.18. Para obtener más información, consulte Versión 1.6.22 de Memcached .	10 de enero de 2024
ElastiCache para Redis OSS admite tamaños de nodos C7gn adicionales	ElastiCache para Redis OSS admite tamaños de nodos C7gn adicionales.	10 de enero de 2024

ElastiCache para Redis OSS ahora admite la creación de cachés sin servidor	Ahora puede crear cachés sin servidor, lo que simplifica la administración de la caché y permite el escalado al instante para dar soporte a las aplicaciones más exigentes. Para obtener más información, consulte Cómo elegir una opción de implementación. Como parte de esta característica, se agregaron nuevos permisos a `ElastiCacheServiceRolePolicy` y `AmazonElastiCacheFullAccess` a fin de permitir la asociación de cachés sin servidor con puntos de conexión de VPC administrados. Además, se agregaron permisos para revisar la experiencia de uso de la consola mediante la política `AmazonElastiCacheFullAccess`.	27 de noviembre de 2023

ElastiCache para Memcached ahora admite la creación de cachés sin servidor	Ahora puede crear cachés sin servidor, lo que simplifica la administración de la caché y permite el escalado al instante para dar soporte a las aplicaciones más exigentes. Para obtener más información, consulte Cómo elegir una opción de implementación. Como parte de esta característica, se agregaron nuevos permisos a `ElastiCacheServiceRolePolicy` y `AmazonElastiCacheFullAccess` a fin de permitir la asociación de cachés sin servidor con puntos de conexión de VPC administrados. Además, se agregaron permisos para revisar la experiencia de uso de la consola mediante la política `AmazonElastiCacheFullAccess` .	27 de noviembre de 2023
ElastiCache para Redis OSS ahora admite la modificación del modo de clúster	Ahora puede migrar los clústeres del modo de clúster desactivado (CMD) al modo de clúster habilitado (CME). Para obtener más información, consulte Modificación del modo de clúster.	11 de mayo de 2023

ElastiCache para Redis OSS ahora admite la modificación del cifrado en tránsito	Ahora puede cambiar la configuración de TLS de los clústeres de Redis OSS sin necesidad de volver a crear o aprovisionar los clústeres ni afectar a la disponibilidad de las aplicaciones. Para obtener más información, consulte Habilitación del cifrado en tránsito en un clúster existente .	28 de diciembre de 2022
ElastiCache para Redis OSS ahora admite la autenticación de usuarios mediante IAM	La autenticación de IAM le permite autenticar una conexión a ElastiCache para Redis OSS mediante identidades de IAM de AWS. Esto le permite reforzar el modelo de seguridad y simplificar muchas tareas de seguridad administrativa. Para obtener más información, consulte Authenticating with IAM (Autenticación con IAM).	16 de noviembre de 2022
ElastiCache para Redis OSS ahora admite Redis OSS 7	Esta versión incorpora varias características nuevas en Amazon ElastiCache para Redis: funciones de Redis OSS, mejoras de ACL y Pub/Sub fragmentado. Para obtener más información, consulte ElastiCache para Redis OSS versión 7.0.	8 de noviembre de 2022

ElastiCache para Redis OSS ahora admite IPV6	ElastiCache admite las versiones 4 y 6 del Protocolo de Internet (IPv4 e IPv6), lo que le permite configura r el clúster para que acepte solo conexiones IPv4, solo conexiones IPv6 o conexione s IPv4 e IPv6 (doble pila). Se admite IPv6 para cargas de trabajo que utilicen la versión 6.2 y posteriores del motor de Redis OSS en todas las instancias creadas en Nitro System. No hay cargos adicionales por acceder a ElastiCache a través de IPv6. Para obtener más información, consulte Choosing a network type (Elección del tipo de red).	7 de noviembre de 2022

ElastiCache for Memcached ahora es compatible con IPV6	ElastiCache admite las versiones 4 y 6 del Protocolo de Internet (IPv4 e IPv6), lo que le permite configura r el clúster para que acepte solo conexiones IPv4, solo conexiones IPv6 o conexione s IPv4 e IPv6 (doble pila). Se admite IPv6 para cargas de trabajo que utilicen la versión 1.6.6 o superior del motor Memcached en todas las instancias creadas en el sistema Nitro. No hay cargos adicionales por acceder a ElastiCache a través de IPv6. Para obtener más información, consulte Choosing a network type (Elección del tipo de red).	7 de noviembre de 2022
ElastiCache para Memcached ahora admite el cifrado en tránsito	El cifrado en tránsito es una característica opcional que permite reforzar la seguridad de los datos en sus momentos más vulnerables: cuando se trasladan de una ubicación a otra. Se admite en las versiones 1.6.12 y posterior es de Memcached. Para obtener más información, consulte Cifrado en tránsito de ElastiCache (TLS).	26 de mayo de 2022

ElastiCache para Redis OSS ahora admite el formato JavaScript Object Notation (JSON) nativo	El formato nativo de JavaScript Object Notation (JSON, Notación de objetos de JavaScript) es una forma sencilla y sin esquemas de codificar conjuntos de datos complejos dentro de clústeres de Redis OSS. Puede almacenar datos de forma nativa y acceder a ellos utilizando el formato JavaScript Object Notation (JSON) dentro de clústeres de Redis OSS, así como actualizar los datos de JSON almacenados en esos clústeres, sin necesidad de administrar un código personalizado para serializarlo y deserializarlo. Para obtener más información, consulte Introducción a JSON.	25 de mayo de 2022

ElastiCache ahora admite PrivateLink	AWS PrivateLink permite acceder de forma privada a las operaciones de la API de ElastiCache sin una puerta de enlace de Internet, un dispositivo NAT, una conexión VPN o una AWS Direct Connect conexión. Para obtener más información, consulte Amazon ElastiCache API and interface VPC endpoints (AWS PrivateLink) para Redis o Amazon ElastiCache API and interface VPC endpoints (AWS PrivateLink) para Memcached.	24 de enero de 2022

ElastiCache para Redis OSS ahora admite Redis OSS 6.2 y la organización de datos en niveles	Amazon ElastiCache para Redis OSS introduce la próxima versión del motor de Redis OSS compatible con Amazon ElastiCache. ElastiCache para Redis OSS 6.2 incluye mejoras de rendimiento para clústeres habilitados para TLS que utilizan tipos de nodos x86 con 8 vCPUs o más o tipos de nodos Graviton2 con 4 vCPUs o más. ElastiCache para Redis OSS también introduce la organización de datos en niveles. Puede utilizar la organización de datos en niveles como una forma más económica de escalar los clústeres hasta cientos de terabytes de capacidad. Para obtener más información, consulte Versión 6.2 de ElastiCache para Redis OSS versión 6.2 (mejorada) y Organización de datos en niveles.	23 de noviembre de 2021

Respaldo para Auto Scaling	ElastiCache para Redis OSS ahora es compatible con el escalado automático. El escalado automático de ElastiCache para Redis permite aumentar o disminuir las particiones o réplicas deseadas en el servicio de ElastiCache para Redis OSS de forma automática. ElastiCache utiliza el servicio de Auto Scaling de aplicaciones para proporcionar esta funcionalidad. Para obtener más información, consulte Escalado automático de clústeres de ElastiCache para Redis OSS.	19 de agosto de 2021
Compatibilidad con la entrega de registros lentos de Redis OSS	ElastiCache ahora permite transmitir SLOWLOG de Redis OSS a uno de dos destinos: Amazon Data Firehose o Registros de Amazon CloudWatch. Para obtener más información, consulte Entrega de registros.	22 de abril de 2021

Respaldo para etiquetado de recursos y claves de condición	ElastiCache ahora admite el etiquetado para ayudarlo a administrar los clústeres y otros recursos de ElastiCache. Para obtener más informaci ón, consulte Etiquetado de los recursos de ElastiCache. ElastiCache también introduce respaldo para las claves de condición. Puede especificar condiciones que determinan cómo se aplica una política de IAM. Para obtener más información, consulte Uso de las claves de condición.	7 de abril de 2021
Respaldo para etiquetado de recursos y claves de condición	ElastiCache ahora admite el etiquetado para ayudarlo a administrar los clústeres y otros recursos de ElastiCache. Para obtener más informaci ón, consulte Etiquetado de los recursos de ElastiCache. ElastiCache también introduce respaldo para las claves de condición. Puede especificar condiciones que determinan cómo se aplica una política de IAM. Para obtener más información, consulte Uso de las claves de condición.	7 de abril de 2021

ElastiCache ya se encuentra disponible en AWS Outposts	AWS Outposts brinda servicios de AWS nativos, infraestructura y modelos operativos a prácticamente cualquier centro de datos, espacio de coubicación o instalación local. Puede implementar ElastiCache en Outposts para configurar, operar y utilizar la caché en las instalaciones, como lo haría en la nube. Para obtener más información, consulte Uso de Outposts.	8 de octubre de 2020
ElastiCache ahora es compatible con Redis OSS 6	Amazon ElastiCache para Redis OSS introduce la próxima versión del motor de Redis OSS compatible con Amazon ElastiCache. Esta versión incluye autenticación de usuarios con control de acceso basado en roles, respaldo sin versiones, almacenamiento en caché del lado del cliente y mejoras operativas significativas. Para obtener más información, consulte ElastiCache para Redis OSS versión 6.0 (mejorada).	7 de octubre de 2020

ElastiCache ahora admite Local Zones	Una Zona local es una extensión de una región de AWS que está geográfic amente cerca de sus usuarios. Puede ampliar cualquier nube virtual privada (VPC) de la región de AWS principal a Local Zones al crear una subred nueva y asignarla a la zona local. Para obtener más información, consulte Uso de Local Zones.	25 de septiembre de 2020
ElastiCache para Redis OSS ahora admite el escalado del entorno del clúster de Redis OSS en hasta 500 nodos o 500 particiones	El modo de clúster de Redis OSS permite configuraciones que puede utilizar para dividir sus datos entre diferentes particiones y ofrece una mejor escalabilidad, rendimiento y disponibilidad. Esta caracterí stica se encuentra disponibl e en Amazon ElastiCache para Redis OSS versión 5.0.6 en adelante en todos las regiones de AWS y para todos los entornos de clúster de ElastiCache para Redis OSS existentes y nuevos. Para obtener más información, consulte Redis OSS Nodes and Shards.	13 de agosto de 2020

ElastiCache ahora es compatible con permisos de nivel de recursos	Puede restringir el alcance de los permisos de un usuario mediante la especificación de recursos de ElastiCache en una política de AWS Identity and Access Managemen t (IAM). Para obtener más información, consulte los permisos de nivel de recursos.	12 de agosto de 2020
ElastiCache para Redis OSS añade métricas adicionales de Amazon CloudWatch	ElastiCache para Redis OSS ahora admite métricas de CloudWatch nuevas, incluido `PubSubCmds` y `HyperLogL ogBasedCmds` . Para obtener una lista completa, consulte Metrics for Redis OSS.	10 de junio de 2020
ElastiCache ahora admite la actualización automática de clústeres de ElastiCache	Amazon ElastiCache ahora admite la actualización automática de los clústeres de ElastiCache después de que haya pasado la “fecha de aplicación recomendada” de la actualización de servicio. ElastiCache utilizará el periodo de mantenimiento para programar la actualización automática de los clústeres aplicables. Para obtener más información, consulte Actualizaciones de autoservi cio.	13 de mayo de 2020

ElastiCache para Redis OSS ahora es compatible con el almacén de datos global de Redis OSS	La característica del almacén de datos global de Redis OSS ofrece una replicación completamente administrada, rápida, fiable y segura en todas las regiones de AWS. Con esta característica, puede crear clústeres de réplicas de lectura entre regiones para que ElastiCache para Redis OSS permita lecturas de baja latencia y recuperación ante desastres en todas las regiones de AWS. Puede crear, modificar y describir un almacén de datos global. También puede agregar o eliminar regiones de AWS de su almacén de datos global y promover una región de AWS como principal dentro de un almacén de datos global. Para obtener más información, consulte Replicación en regiones de AWS mediante Global Datastore.	16 de marzo de 2020
ElastiCache para Redis OSS ahora es compatible con la versión 5.0.6 de Redis OSS	Para obtener más información, consulte ElastiCache para Redis OSS versión 5.0.6 (mejorada).	18 de diciembre de 2019

Amazon ElastiCache ahora admite nodos de caché T3 estándar	Ahora puede lanzar la próxima generación de nodos de caché T3 estándar ampliables de uso general en Amazon ElastiCache. Las instancias T3 estándar de Amazon EC2 proporcionan un nivel básico de rendimiento de la CPU con la capacidad de explotar la utilización de la CPU en cualquier momento hasta que se agoten los créditos acumulados. Para obtener más información, consulte Tipos de nodos compatibles.	12 de noviembre de 2019

Amazon ElastiCache ahora admite la modificación del token AUTH en un servidor de ElastiCache para Redis OSS existente	ElastiCache para Redis OSS 5.0.6 ahora permite modificar los tokens de autenticación al configura r y rotar los tokens nuevos. Puede modificar los tokens activos mientras están en uso. También puede añadir tokens nuevos a los clústeres habilitados con cifrado en tránsito que se configuraron anteriormente sin tokens de autenticación. Se trata de un proceso de dos pasos mediante el cual puede configurar y rotar el token sin interrumpir las solicitudes de los clientes. Esta característica no se admite actualmente en CloudFormation. Para obtener más información, consulte Authenticating Users with the Redis OSS AUTH Command.	30 de octubre de 2019
Amazon ElastiCache ahora admite la migración de datos en línea desde Redis OSS en Amazon EC2	Ahora puede utilizar la migración en línea para migrar sus datos desde Redis OSS autoalojado en Amazon EC2 a Amazon ElastiCache. Para obtener más información consulte Migración en línea a ElastiCache.	28 de octubre de 2019

ElastiCache para Redis OSS introduce el escalado vertical en línea para el modo clúster de Redis OSS.	Ahora puede escalar o reducir verticalmente el tamaño de su clúster de Redis OSS particionado bajo demanda. ElastiCache para Redis OSS cambia el tamaño del clúster al modificar el tipo de nodo, a la vez que el clúster permanece en línea y atiende solicitudes de entrada. Para obtener más información, consulte Escalado vertical en línea mediante la modificación del tipo de nodo.	20 de agosto de 2019
ElastiCache para Redis OSS ahora permite a los usuarios utilizar un solo punto de conexión del lector para el clúster de Amazon ElastiCache para Redis OSS.	Esta característica le permite dirigir todo el tráfico de lectura a su clúster de ElastiCache para Redis OSS a través de un único punto de conexión en el clúster para aprovechar el equilibrador de carga y conseguir una mayor disponibilidad. Para obtener más información, consulte Búsqueda de puntos de enlace de conexión.	13 de junio de 2019

ElastiCache para Redis OSS ahora permite a los usuarios aplicar las actualizaciones de servicio de acuerdo con su propia programación	Con esta característica, puede elegir aplicar las actualizaciones del servicio en el momento que lo desee y no solo durante los periodos de mantenimiento. De esta forma, minimizará las interrupciones del servicio, especialmente durante los picos de flujo de trabajo, y se asegurará de cumplir los requisitos de conformidad si su clúster está en un programa de conformidad compatible con ElastiCache. Para obtener más información, consulte Self-Service Updates in Amazon ElastiCache y Compliance validation for Amazon ElastiCache.	4 de junio de 2019
Ofertas de instancias reservadas estándar de ElastiCache: sin pago inicial, pago inicial parcial y pago total anticipado.	Las instancias reservadas ofrecen la flexibilidad necesaria para reservar una instancia de Amazon ElastiCache durante un plazo de uno a tres años en función de la región de AWS y el tipo de instancias. Para obtener más información, consulte Administración de costos con los nodos reservados.	18 de enero de 2019

ElastiCache para Redis OSS admite hasta 250 nodos por clúster de Redis OSS	El límite de nodos o de particiones puede aumentarse hasta un máximo de 250 por clúster de ElastiCache para Redis OSS. Para obtener más información, consulte Particiones.	19 de noviembre de 2018
ElastiCache para Redis OSS es compatible con la conmutación por error automática y la copia de seguridad y restauración en todos los nodos T2	ElastiCache para Redis OSS ahora es compatible con la conmutación por error automática, la creación de instantáneas y la copia de seguridad y restauración en todos los nodos T2. Para obtener más información, consulte Copia de seguridad y restauración de ElastiCache para Redis OSS e Instantánea.	19 de noviembre de 2018
Compatibilidad con ElastiCache para Redis OSS para nodos M5 y R5	ElastiCache para Redis OSS ahora es compatible con los nodos R5 y M5, que son tipos de instancias optimizadas para memoria y de uso general basadas en el sistema AWS Nitro System. Para obtener más información, consulte Tipos de nodos compatibles.	23 de octubre de 2018

Posibilidad de cambiar dinámicamente el número de réplicas de lectura	ElastiCache para Redis OSS incluye la posibilidad de añadir y eliminar réplicas de lectura de cualquier clúster sin tiempo de inactividad del clúster. Para obtener más información sobre este y otros cambios de esta versión, consulte Cambio del número de réplicas en la Guía del usuario de ElastiCache para Redis. Consulte también DecreaseReplicaCount e IncreaseReplicaCount en la Referencia de la API de ElastiCache.	17 de septiembre de 2018
Certificación de conformidad con FedRAMP	ElastiCache para Redis OSS ha obtenido la certificación de conformidad con FedRAMP. Para obtener más información, consulte Compliance validation for Amazon ElastiCache.	30 de agosto de 2018
Actualizaciones de motores de Valkey o Redis OSS (modo de clúster habilitado	Amazon ElastiCache para Redis OSS ha añadido compatibilidad con la actualización de versiones del motor de Valkey o Redis OSS (modo de clúster habilitado). Para obtener más información, consulte Actualización de las versiones del motor.	20 de agosto de 2018

Certificación de conformidad con PCI DSS	ElastiCache para Redis OSS ha obtenido la certificación de conformidad con PCI DSS. Para obtener más información, consulte Compliance validation for Amazon ElastiCache.	5 de julio de 2018
Compatibilidad con ElastiCache para Redis OSS 4.0.10	ElastiCache para Redis OSS ahora admite Redis OSS 4.0.10, incluido el cifrado y el cambio de tamaño de clústeres en línea en una sola versión. Para obtener más información, consulte ElastiCache para Redis OSS versión 4.0.10 (mejorada).	14 de junio de 2018
Estructura de la guía del usuario	La Guía del usuario de ElastiCache única se ha reestructurado a fin de que haya guías de usuario separadas para Redis OSS (Guía del usuario de ElastiCache para Redis) y Memcached (Guía del usuario de ElastiCache para Memcached). La estructura de la documentación en la sección de la Referencia de comandos de la AWS CLI: elasticache y en la Referencia de la API de Amazon ElastiCache permanece igual.	20 de abril de 2018

Respaldo con la métrica EngineCPUUtilization	ElastiCache para Redis OSS incluye una nueva métrica, `EngineCPUUtilization`, que notifica el porcentaje de la capacidad de la CPU que se utiliza actualmente. Para obtener más información, consulte Metrics for Redis OSS.	9 de abril de 2018

En la siguiente tabla, se describen los cambios importantes en la Guía del usuario de ElastiCache anteriores a marzo de 2018.

Cambio	Descripción	Fecha de modificación
Soporte para la región Asia-Pacífico (Osaka-Local).	ElastiCache ha agregado soporte para la región Asia-Pacífico (Osaka-Local). La región Asia-Pacífico (Osaka) admite actualmente una única zona de disponibilidad y solo es por invitación. Para obtener más información, consulte los siguientes temas: • Regiones admitidas • Tipos de nodos de caché compatibles	12 de febrero de 2018
Soporte para EU (París)	ElastiCache ha agregado soporte para la región EU (París). Para obtener más información, consulte los siguientes temas: • Regiones admitidas • Tipos de nodos de caché compatibles	18 de diciembre de 2017

Cambio	Descripción	Fecha de modificación
Soporte para la región China (Ningxia)	Amazon ElastiCache ha agregado soporte para la región China (Ningxia) Para obtener más información, consulte los siguientes temas: • Regiones admitidas • Tipos de nodos de caché compatibles	11 de diciembre de 2017
Respaldo con para roles vinculados a servicios	Esta versión de ElastiCache ha agregado soporte para los roles vinculados a servicios (SLR). Para obtener más información, consulte los siguientes temas: • Uso de roles vinculados a servicios para Amazon ElastiCache • Configurar los permisos (solo para nuevos usuarios de ElastiCache)	7 de diciembre de 2017
Respaldo con los tipos de nodo R4	Esta versión de ElastiCache ha agregado soporte para los tipos de nodo R4 en todas las regiones de AWS compatibles con ElastiCache. Puede adquirir nodos del tipo R4 como nodos de caché reservados o en diferido. Para obtener más información, consulte los siguientes temas: • Tipos de nodos de caché compatibles • Parámetros específicos de tipo de nodo de Memcached • Parámetros específicos de tipos de nodos de Redis OSS	20 de noviembre de 2017

Cambio	Descripción	Fecha de modificación
ElastiCache para Redis OSS 3.2.10 y compatibilidad con el cambio de las particiones en línea	Amazon ElastiCache para Redis OSS añade compatibilidad con ElastiCache para Redis 3.2.10. ElastiCache para Redis OSS también presenta el cambio de tamaño de clúster en línea para añadir o eliminar particiones del clúster mientras sigue atendiendo solicitudes de E/S entrantes. Para obtener más información, consulte los siguientes temas: • Cambio de tamaño de clústeres online • Cambio de particiones en línea para Valkey o Redis OSS (modo de clúster habilitado)	9 de noviembre de 2017
Conformidad con HIPAA	La versión 3.2.6 de ElastiCache para Redis OSS ya tiene certificación para la conformidad con HIPAA cuando el cifrado se encuentra habilitad o en el clúster. Para obtener más información, consulte los siguientes temas: • Validación de la conformidad para Amazon ElastiCache • Seguridad de datos en Amazon ElastiCache	2 de noviembre de 2017

Cambio	Descripción	Fecha de modificación
ElastiCache para Redis OSS 3.2.6 y compatibilidad con el cifrado	ElastiCache añade compatibilidad con ElastiCache para Redis OSS 3.2.6, que incluye dos caracterí sticas de cifrado: • El cifrado en tránsito cifra los datos mientras están en tránsito; por ejemplo, entre los nodos de un clúster o entre un clúster y la aplicación. • El cifrado en reposo cifra los datos en el disco durante las operaciones de sincronización y copia de seguridad. Para obtener más información, consulte los siguientes temas: • Seguridad de datos en Amazon ElastiCache • Motores y versiones compatibles	25 de octubre de 2017
Tema de patrones de conexión	En la documentación de ElastiCache se agrega un tema que abarca varios patrones para el acceso a un clúster de ElastiCache en una Amazon VPC. Para obtener más información, consulte Patrones de acceso para obtener acceso a una caché de ElastiCache en una Amazon VPC en la Guía del usuario de ElastiCache.	24 de abril de 2017
Soporte para Memcached 1.4.34	ElastiCache agrega soporte para la versión 1.4.34 de Memcached, que incorpora una serie de correcciones de versiones anteriores de Memcached. Para obtener más información, consulte Notas de la versión de Memcached 1.4.34 en Memcached en GitHub.	10 de abril de 2017

Cambio	Descripción	Fecha de modificación
Posibilidad de realizar pruebas con la conmutación por error automática	ElastiCache añade compatibilidad para la realización de pruebas de conmutación por error automática en clústeres de Redis OSS que admiten la replicación. Para obtener más información, consulte los siguientes temas: • Prueba de la conmutación por error automática en la Guía del usuario de ElastiCache. • TestFailover en la Referencia API de ElastiCache . • test-failover en la Referencia de la AWS CLI.	4 de abril de 2017
Mejoras en la restauración de Redis OSS	ElastiCache añade mejoras en las funciones de copia de seguridad y restauración de Redis OSS con el cambio de tamaño de clústeres. Esta característica permite restaurar una copia de seguridad en otro clúster con un número de particiones distinto del que tenía el clúster que se utilizó para crear la copia de seguridad. (Para la API y la CLI, esta característica puede restaurar un número distinto de grupos de nodos en lugar de un número distinto de particiones). Esta actualización también admite diferentes configuraciones de ranuras de Redis OSS. Para obtener más información, consulte Restauración desde una copia de seguridad a una nueva caché.	15 de marzo de 2017

Cambio	Descripción	Fecha de modificación
Nuevo parámetro de administración de memoria de Redis OSS	ElastiCache añade un nuevo parámetro de Redis OSS, `reserved-memory-percent` , que facilita la administración de la memoria reservada . Este parámetro se encuentra disponible en todas las versiones de ElastiCache para Redis OSS. Para obtener más información, consulte los siguientes temas: • Administración de la memoria reservada para Valkey y Redis OSS • Parámetros nuevos para Redis OSS 3.2.4	15 de marzo de 2017
Soporte para Memcached 1.4.33	ElastiCache agrega soporte para la versión 1.4.33 de Memcached. Para obtener más información, consulte los siguientes temas: • ElastiCache versión 1.4.33 para Memcached • Parámetros agregados a Memcached 1.4.33	20 de diciembre de 2016
Respaldo con la región UE Oeste (Londres)	ElastiCache agrega soporte para la región EU (Londres). Actualmente, solo se admiten los tipos de nodos T2 y M4. Para obtener más información, consulte los siguientes temas: • Regiones admitidas • Tipos de nodos de caché compatibles	13 de diciembre de 2016
Respaldo con la región Canadá (Montreal)	ElastiCache agrega soporte para la región Canadá (Montreal). Actualmente, en esta región de AWS solo se admiten los tipos de nodo M4 y T2. Para obtener más información, consulte los siguientes temas: • Regiones admitidas • Tipos de nodos de caché compatibles	8 de diciembre de 2016

Cambio	Descripción	Fecha de modificación
Respaldo con los tipos de nodo M4 y R3	ElastiCache agrega soporte para los tipos de nodo R3 y M4 en la región América del Sur (São Paulo) y con los tipos de nodo M4 en la región China (Pekín). Para obtener más información, consulte los siguientes temas: • Regiones admitidas • Tipos de nodos de caché compatibles	1 de noviembre de 2016
Respaldo con la región EE. UU Este 2 (Ohio)	ElastiCache agrega soporte para la región EE. UU. Este (Ohio) (us-east-2) con los tipos de nodo M4, T2 y R3. Para obtener más información, consulte los siguientes temas: • Regiones admitidas • Tipos de nodos de caché compatibles	17 de octubre de 2016

Cambio	Descripción	Fecha de modificación
Compatibilidad con clústeres de Redis OSS	ElastiCache añade compatibilidad con los clústeres de Redis OSS (mejorado). Los clientes que utilicen clústeres de Redis OSS, podrán particionar sus datos en hasta 15 particiones (grupos de nodos). Cada partición admite la reproducción con hasta 5 réplicas de lectura por partición. Los tiempos de conmutación por error automática de los clústeres de Redis OSS son de tan solo una cuarta parte del tiempo que necesitab an las versiones anteriores. Esta versión incluye una consola de administración rediseñada que utiliza terminología en consonanc ia con el uso del sector. Para obtener más información, consulte los siguientes temas: • Comparación entre Memcached y Redis OSS • Componentes y características de ElastiCache: tenga en cuenta las secciones sobre los nodos, las particiones, los clústeres y la reproducción. • Terminología de ElastiCache	12 de octubre de 2016

Cambio	Descripción	Fecha de modificación
Respaldo con el tipo de nodo M4	ElastiCache agrega soporte para la familia de tipos de nodo M4 en la mayoría de las regiones de AWS compatibles con ElastiCache. Puede adquirir nodos del tipo M4 como nodos de caché reservados o en diferido. Para obtener más información, consulte los siguientes temas: • Tipos de nodos de caché compatibles • Parámetros específicos de tipo de nodo de Memcached • Parámetros específicos de tipos de nodos de Redis OSS	3 de agosto de 2016
Respaldo con la región Mumbai	ElastiCache agrega soporte para la región Asia-Pacífico (Mumbai). Para obtener más información, consulte los siguientes temas: • Tipos de nodos de caché compatibles • Parámetros específicos de tipo de nodo de Memcached • Parámetros específicos de tipos de nodos de Redis OSS	27 de junio de 2016
Exportación de instantáneas	ElastiCache añade la posibilidad de exportar una instantánea de Redis OSS para que pueda obtener acceso a ella desde fuera de ElastiCache. Para obtener más información, consulte los siguientes temas: • Exportación de una copia de seguridad en la Guía del usuario de Amazon ElastiCache • CopySnapshot en la Referencia de la API de Amazon ElastiCache	26 de mayo de 2016

Cambio	Descripción	Fecha de modificación
Escalado vertical del tipo de nodo	ElastiCache añade la posibilidad de escalar verticalmente el tipo de nodo de Redis OSS. Para obtener más información, consulte Escalado de ElastiCache.	24 de marzo de 2016
Simplificación de la actualización del motor	ElastiCache añade la posibilidad de actualizar fácilmente el motor de caché de Redis OSS. Para obtener más información, consulte Administración de versiones para ElastiCache.	22 de marzo de 2016
Respaldo con los tipos de nodo R3	ElastiCache agrega soporte para los tipos de nodo R3 en las regiones China (Pekín) y América del Sur (São Paulo). Para obtener más información, consulte Tipos de nodos de caché compatibles.	16 de marzo de 2016
Acceso a ElastiCache mediante una función de Lambda	Se ha agregado un tutorial acerca de cómo configurar una función Lambda para tener acceso a ElastiCache en una VPC de Amazon. Para obtener más información, consulte Otros tutoriales y vídeos de ElastiCache.	12 de febrero de 2016
Compatibilidad con Redis OSS 2.8.24	ElastiCache añade compatibilidad con la versión 2.8.24 de Redis OSS con mejoras incorporadas a partir de Redis OSS 2.8.23. Entre las mejoras se incluyen varias correcciones de errores y respaldo con el registro de direcciones de acceso a la memoria incorrectas. Para obtener más información, consulte los siguientes temas: • Versión 2.8.24 de ElastiCache para Redis OSS (mejorada) • Redis OSS 2,8 Release Notes	20 de enero de 2016
Soporte para la región Asia-Pacífico (Seúl).	ElastiCache agrega soporte para la región Asia-Pacífico (Seúl) (ap-northeast-2) con los tipos de nodos t2, m3 y r3.	6 de enero de 2016

Cambio	Descripción	Fecha de modificación
Cambio de consola de Amazon ElastiCache.	Dado que las versiones de Redis OSS más recientes ofrecen una experiencia del usuario mejor y más estable, las versiones 2.6.13, 2.8.6 y 2.8.19 de Redis OSS ya no se incluyen en la consola de administración de ElastiCache. Para obtener más opciones y más información, consulte Motores y versiones compatibles.	15 de diciembre de 2015
Compatibilidad con Redis OSS 2.8.23.	ElastiCache añade compatibilidad con la versión 2.8.23 de Redis OSS con mejoras incorporadas a partir de Redis OSS 2.8.22. Entre las mejoras se incluye corrección de errores y respaldo con el nuevo parámetro `close-on-slave-write` que, si está habilitado, desconecta a los clientes que intentan escribir en una réplica de solo lectura. Para obtener más información, consulte Versión 2.8.23 de ElastiCache para Redis OSS (mejorada).	13 de noviembre de 2015

Cambio	Descripción	Fecha de modificación
Compatibilidad con Redis OSS 2.8.22.	ElastiCache añade compatibilidad con la versión 2.8.22 de Redis OSS con mejoras incorporadas de ElastiCache a partir de la versión 2.8.21. Entre las mejoras, se incluyen las características siguientes: • Implementación de un proceso de almacenamiento sin ramificaciones que permite almacenar contenido correctamente cuando una situación de escasa memoria disponible podría provocar errores en la operación de almacenamiento con ramificaciones. • Métricas de CloudWatch adicionales: SaveInProgress y ReplicationBytes. • Con el fin de habilitar las sincronizaciones parciales, el parámetro `repl-backlog-size` de Redis OSS ahora se aplica a todos los clústeres. Para obtener una lista completa de los cambios, así como información adicional, consulte Versión 2.8.22 de ElastiCache para Redis OSS (mejorada). Esta versión de la documentación incluye una reorganización de la documentación y ha eliminado la documentación de la interfaz de línea de comandos (CLI) de ElastiCache. A fin de obtener sobre información sobre el uso de la línea de comandos, consulte Línea de comandos de AWS para ElastiCache.	28 de septiembre de 2015

Cambio	Descripción	Fecha de modificación
Soporte para Memcached 1.4.28	ElastiCache agrega soporte para la versión 1.4.24 de Memcached y con las mejoras de Memcached incorporadas a partir de la versión 1.4.14. Esta versión permite administrar la caché de los elementos menos usados recientemente (LRU) como tarea en segundo plano y elegir jenkins o murmur3 como algoritmos de hash; asimismo, incluye nuevos comandos y varias correcciones de errores. Para obtener más información, consulte las notas de la versión de Memcached. .	27 de agosto de 2015
Soporte para la detección automática de Memcached con PHP 5.6	Esta versión de Amazon ElastiCache agrega soporte para los clientes de detección automática de Memcached para PHP 5.6. Para obtener más información, consulte Compilación del código fuente de ElastiCache Cluster Client para PHP.	29 de julio de 2015
Compatibilidad con Redis OSS 2.8.21	ElastiCache añade compatibilidad con la versión 2.8.21 de Redis OSS y mejoras de Redis OSS incorporadas a partir de la versión 2.8.19. Esta versión de Redis OSS incluye varias correcciones de errores. Para obtener más información, consulte Redis OSS 2.8 release notes.	29 de julio de 2015
Nuevo tema: acceso a ElastiCache desde fuera de AWS	Se ha agregado un nuevo tema que describe cómo obtener acceso a los recursos de ElastiCache desde fuera de AWS. Para obtener más información, consulte Acceso a recursos de ElastiCache desde fuera de AWS.	9 de julio de 2015

Cambio	Descripción	Fecha de modificación
Incorporación de mensajes de reemplazo de nodos	ElastiCache agrega tres mensajes relacionados con la sustitución programada de nodos: ElastiCache:NodeReplacementScheduled, ElastiCache:NodeReplacementRescheduled y ElastiCache:NodeReplacementCanceled. Para obtener más información y detalles acerca de las acciones que puede realizar cuando se programa el reemplazo de un nodo, consulte de ElastiCache Notificaciones de eventos y Amazon SNS.	11 de junio de 2015

Cambio	Descripción	Fecha de modificación
Compatibilidad con Redis OSS 2.8.19.	ElastiCache añade compatibilidad con la versión 2.8.19 de Redis OSS y mejoras de Redis OSS incorporadas a partir de la versión 2.8.6. Este apoyo incluye respaldo con: • La estructura de datos de HyperLogLog, con los comandos de Redi OSS PFADD, PFCOUNT y PFMERGE. • Las consultas de rango lexicográfico con los nuevos comandos ZRANGEBYLEX, ZLEXCOUNT y ZREMRANGEBYLEX. • Se han introducido una serie de correcciones de errores que impiden que un nodo primario envíe datos obsoletos a nodos de réplica. Esto se consigue mediante un error en la operación SYNC principal cuando un proceso secundario de almacenamiento en segundo plano (bgsave) termina de forma inesperada. Para obtener más información acerca de HyperLogLog, consulte Redis OSS new data structure: the HyperLogLog. Para obtener más información acerca de PFADD, PFCOUNT y PFMERGE, consulte la documentación de Redis OSS y haga clic en HyperLogLog.	11 de marzo de 2015
Respaldo con las etiquetas de asignación de costos	ElastiCache agrega soporte para las etiquetas de asignación de costos. Para obtener más información, consulte Monitoreo de costos con etiquetas de asignación de costos.	9 de febrero de 2015

Cambio	Descripción	Fecha de modificación
Respaldo para la región AWS GovCloud (EE. UU. Oeste)	ElastiCache agrega soporte para la región AWS GovCloud (EE. UU. Oeste) (us-gov-west-1).	29 de enero de 2015
Soporte para la región de Europa (Fráncfort)	ElastiCache agrega soporte para la región de Europa (Fráncfort) (eu-central-1).	19 de enero de 2015
Compatibilidad con multi-AZ para grupos de replicación de Redis OSS	ElastiCache añade compatibilidad con multi-AZ desde el nodo principal a la réplica de lectura del grupo de replicación de Redis OSS. ElastiCac he monitorea el estado del grupo de reproducc ión. Si se produce un error en el nodo primario, ElastiCache promociona automáticamente una réplica a nodo primario y, a continuación, reemplaza la réplica. Para obtener más informaci ón, consulte Minimización del tiempo de inactivid ad en ElastiCache utilizando multi-AZ con Valkey y Redis OSS.	24 de octubre de 2014
AWS CloudTrai lRespaldo con el registro de llamadas a la API de	ElastiCache agrega soporte para el uso de AWS CloudTrail a fin de registrar todas las llamadas a la API de ElastiCache. Para obtener más informaci ón, consulte Registro de llamadas a la API de Amazon ElastiCache con AWS CloudTrail.	15 de septiembre de 2014
Respaldo con nuevos tamaños de instancias	ElastiCache agrega soporte para las instancia s de uso general (T2) adicionales. Para obtener más información, consulte Configuración de los parámetros de motor mediante grupos de parámetros de ElastiCache.	11 de septiembre de 2014

Cambio	Descripción	Fecha de modificación
Respaldo con la ubicación de nodos flexible en Memcached	ElastiCache agrega soporte para crear nodos de Memcached en varias zonas de disponibilidad.	23 de julio de 2014
Respaldo con nuevos tamaños de instancias	ElastiCache agrega soporte para las instancia s de uso general (M3) adicionales e instancias optimizadas para memoria (R3). Para obtener más información, consulte Configuración de los parámetros de motor mediante grupos de parámetros de ElastiCache.	1 de julio de 2014
Detección automática PHP	Se ha agregado respaldo con la detección automática de PHP, versión 5.5.	13 de mayo de 2014
Copia de seguridad y restauración para clústeres de Redis OSS	En esta versión, ElastiCache permite a los clientes crear instantáneas de sus clústeres de Redis OSS y crear nuevos clústeres con dichas instantán eas. Una copia de seguridad es una copia del clúster realizada en un momento determinado y se compone de metadatos de clústeres y de todos los datos de la caché de Redis OSS. Las copias de seguridad se almacenan en Amazon S3 y los clientes pueden restaurar los datos a partir de una instantánea en un nuevo clúster en cualquier momento. Para obtener más información, consulte Instantánea y restauración.	24 de abril de 2014

Cambio	Descripción	Fecha de modificación
Redis OSS 2.8.6	ElastiCache es compatible con Redis OSS 2.8.6, además de con Redis OSS 2.6.13. Con Redis OSS 2.8.6, los clientes pueden mejorar la agilidad y la tolerancia a errores de las réplicas de lectura, con posibilidad de realizar resincronización parcial y un número mínimo definido por el usuario de réplicas de lectura que deben estar disponibles en todo momento. Redis OSS 2.8.6 también ofrece compatibilidad completa con características de publicación y suscripción, con las que los clientes pueden recibir notificaciones de eventos que se producen en el servidor.	13 de marzo de 2014

Cambio	Descripción	Fecha de modificación
Motor de la caché de Redis OSS	ElastiCache incluye el software del motor de la caché de Redis OSS, además de Memcached. Los clientes que utilicen Redis OSS actualmente pueden propagar un nuevo clúster de caché de ElastiCache Redis OSS con los datos existentes de archivos de instantáneas de Redis OSS, lo que facilita la migración a un entorno de ElastiCache administrado. Con el fin de ofrecer compatibilidad con las capacidades de replicación de Redis OSS, la API de ElastiCache ahora admite los grupos de replicación. Los clientes pueden crear un grupo de replicación con un nodo de caché de Redis OSS principal y añadir uno o varios nodos de réplica de lectura que se mantienen sincroniz ados automáticamente con los datos de la caché del nodo principal. Las aplicaciones que realizan un uso intensivo de las operaciones de lectura pueden descargarse a una réplica de lectura, lo que reduce la carga en el nodo primario. Las réplicas de lectura también protegen frente a la pérdida de datos en caso de un posible error de nodo de caché primario.	3 de septiembre de 2013
Soporte para Amazon Virtual Private Cloud (VPC) predeterm inado	En esta versión, ElastiCache se encuentra totalmente integrado con la Amazon Virtual Private Cloud (VPC). Para los clientes nuevos, los clústeres se crean en una Amazon VPC de forma predeterminada. Para obtener más informaci ón, consulte Seguridad de ElastiCache y Amazon VPC.	8 de enero de 2013

Cambio	Descripción	Fecha de modificación
Respaldo de PHP con la detección automática de nodos de caché	La versión inicial de la característica de detección automática de nodos de caché ofrecía respaldo con programas de Java. En esta versión, ElastiCache ofrece la característica de detección automática de nodos de caché en PHP.	2 de enero de 2013
Soporte para Amazon Virtual Private Cloud (VPC)	En esta versión, los clústeres de ElastiCache se pueden lanzar en Amazon Virtual Private Cloud (VPC). De forma predeterminada, los clústeres de los clientes nuevos se crean de forma automática en una Amazon VPC. De este modo, los clientes existentes pueden realizar la migración a Amazon VPC a su propio ritmo. Para obtener más información, consulte Seguridad de ElastiCache y Amazon VPC.	20 de diciembre de 2012
Detección automática de nodos de caché y nueva versión de motor de caché	ElastiCache permite utilizar la característica de detección automática de nodos de caché. Esta capacidad permite a los programas de cliente determinar de forma automática todos los nodos de caché de un clúster e iniciar y mantener las conexiones con todos esos nodos. Esta versión también ofrece una nueva versión del motor de caché: la versión 1.4.14 de Memcached. Este nuevo motor de caché ofrece capacidades mejoradas de reequilibrado de slabs, importantes mejoras de rendimiento y escalabilidad, así como varias correcciones de errores. Existen varios nuevos parámetros de caché que pueden configurarse. Para obtener más información, consulte Configuración de los parámetros de motor mediante grupos de parámetros de ElastiCache.	28 de noviembre de 2012

Cambio	Descripción	Fecha de modificación
Nuevos tipos de nodo de caché	Esta versión ofrece cuatro tipos de nodos de caché adicionales.	13 de noviembre de 2012
Nodos de caché reservados	Esta versión ofrece respaldo con los nodos de caché reservados. q	5 de abril de 2012
Nueva guía	Esta es la primera versión de la Guía del usuario de Amazon ElastiCache.	22 de agosto de 2011

AWSGlosario de

Para ver la terminología más reciente de AWS, consulte el Glosario de AWS en la Referencia de Glosario de AWS.